

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

# **DIPLOMOVÁ PRÁCE**

**2010**

**Bc. Tomáš Vondráček**

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Vizualizace informací nad  
mapovým podkladem v MS  
Silverlight**

**Visualization of Map Based  
Information with MS Silverlight**

**2010**

**Bc. Tomáš Vondráček**

# Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě, dne 6. května 2010

Podpis:.....

# Poděkování

Rád bych touto formou poděkoval svému vedoucímu diplomové práce, panu Ing. Michalovi Radeckému, za řadu podnětných nápadů a rad, odborné vedení a za čas, který mi během vypracovávání této práce věnoval.

# Abstrakt

Cílem této práce je zmapování možností vizualizace informací nad mapovým podkladem prostřednictvím technologie MS Silverlight. Práce se zabývá vytvořením testovací aplikace, která ilustruje možnosti integrace a využití mapové komponenty. Na základě zjištěných informací je vytvořena vlastní mapová komponenta, která je rozšířena především o možnosti práce s dalšími informačními vrstvami a formáty.

Aplikace je implementována v prostředí Silverlight. Obsahem práce je specifikace požadavků, analýza a popis samotné implementace. Součástí práce je také instalační a programátorská příručka.

## Klíčová slova:

**Maps, Silverlight, mapová komponenta, ShapeFile, XML, Virtual Earth Silverlight Control, bod zájmu, lomená čára, mnohoúhelník**

# Abstract

The aim of this work is to map out the visualization possibilities of map based information with MS Silverlight. This work deals with creating a test application that illustrates the possibilities of integration and use map components. Based on this informations is created a mapping component that extends the functionality of the above options work with other information layers and formats.

Application is implemented in Silverlight. The content of this thesis is a requirements specification, analysis and description of the implemlementation. The work also includes installation and programming guide.

## Key words:

**Maps, Silverlight, mapping component, ShapeFile, XML, Virtual Earth Silverlight Control, point of interest, polyline, polygon**

# Seznam použitých symbolů a zkratek

<b>AJAX</b>	- Asynchronous JavaScript and XML
<b>ESRI</b>	- Environmental Systems Research Institute
<b>GIS</b>	- Geografický informační systém
<b>GML</b>	- Geography Markup Language
<b>GPU</b>	- Global Processing Unit
<b>GUI</b>	- Graphical user interface
<b>KML</b>	- Keyhole Markup Language
<b>MVC</b>	- Model-view-controller
<b>OGC</b>	- Open Geospatial Consortium
<b>POI</b>	- Point of interest
<b>SDK</b>	- Software development kit
<b>SHP</b>	- Shape File
<b>RIA</b>	- Rich Internet application
<b>UI</b>	- User interface
<b>WFS</b>	- Web Feature Service
<b>WMS</b>	- Web Map Service
<b>WPF</b>	- Windows Presentation Foundation
<b>XAML</b>	- Extensible Application Markup Language
<b>XML</b>	- Extensible Markup Language

# Obsah

1.	Úvod.....	1
1.1	Historie kartografie .....	1
1.2	Kartografie v současnosti a vliv na člověka .....	2
2.	Elektronické mapy .....	3
2.1	Příklady využití zobrazení dat na mapovém podkladu v reálných aplikacích.....	3
2.2	Souřadnicový systém elektronických map .....	6
3.	Vizualizační mapová komponenta pro prostředí MS Silverlight .....	9
3.1	Požadavky na testovací prostředí a mapovou komponentu .....	9
3.1.1	Požadavky na nastavení základní konfigurace .....	9
3.1.2	Požadavky na práci s informačními vrstvami nad daty .....	10
3.2	Možnosti práce s mapou v prostředí MS Silverlight .....	10
3.2.1	BruTile – GIS tiling library .....	10
3.2.2	Map Suite 3.0 Silverlight Edition .....	10
3.2.3	ArcGIS API for Microsoft Silverlight.....	11
3.2.4	Virtual Earth Silverlight Map Control .....	11
4.	Typy geografických dat zobrazených na mapě.....	14
4.1	Bod zájmu (point of interest).....	14
4.2	Lomená čára (polyline).....	15
4.3	Mnohoúhelník (polygon).....	15
5.	Ukládání geografických dat .....	16
5.1	Soubor typu ShapeFile .....	16
5.2	Soubor typu KML .....	19
6.	Optimalizace zobrazených dat .....	24
6.1	Rozdělení obrazovky na regiony .....	24
6.2	Optimalizace rychlosti práce s větším počtem bodů záznamů (dat) .....	25
6.2.1	Proměnné ovlivňující způsob vykreslení.....	28
6.2.2	Případy optimalizace velkého počtu záznamů a vrcholů .....	29
7.	Struktura aplikace.....	34
8.	Implementace .....	35
8.1	Technologie Silverlight .....	35

8.1.1	RadControls for Silverlight .....	37
8.2	Uživatelské rozhraní.....	38
8.3	Rozložení funkcí mapové komponenty .....	39
9.	Budoucí nasazení, rozšíření, využití .....	41
10.	Závěr .....	43
	Literatura .....	44
	Přílohy.....	45



# Seznam obrázků

Obrázek 1: Mapa epidemie dle Johna Snow.....	1
Obrázek 2: Zobrazení aktuálního stavu na silnicích v projektu Floreon.....	3
Obrázek 3: Sledování situace na vodních tocích v projektu Floreon.....	4
Obrázek 4: Zobrazení nemovitostí na prodej na realitním serveru Mapa Nemovitostí.....	4
Obrázek 5: Fotografie zajímavých míst .....	5
Obrázek 6: Sledování polohy přátel.....	5
Obrázek 7: Zaznamenaná trasa v Google Maps .....	6
Obrázek 8: Zobrazení míst, kde je možné si nechat opravit obuv .....	6
Obrázek 9: Souřadnice VŠB-TU Ostrava.....	7
Obrázek 10: Souřadnicové osy .....	8
Obrázek 12: Zobrazení bodu zájmu na mapovém podkladu .....	14
Obrázek 13: Zobrazení lomené čáry na mapovém podkladu .....	15
Obrázek 14: Zobrazení mnohoúhelníku na mapovém podkladu .....	15
Obrázek 15: Struktura hlavního souboru (SHP) .....	17
Obrázek 16: Obrázek s popiskem .....	21
Obrázek 17: Bublina s textem.....	21
Obrázek 18: Mnohoúhelník.....	23
Obrázek 19: Rozdělení obrazovky na regiony.....	24
Obrázek 20: Vykreslení záznamu s vysokým počtem souřadnic .....	25
Obrázek 21: Nevhodně vykreslený mnohoúhelník .....	26
Obrázek 22: Špatně vykreslený mnohoúhelník po přidání pomocných souřadnic .....	26
Obrázek 23: Správně vykreslený mnohoúhelník .....	27
Obrázek 24: Špatně vykreslený mnohoúhelník obrysu ČR.....	27
Obrázek 25: Správně vykreslený mnohoúhelník obrysu ČR.....	28
Obrázek 26: Zobrazení každé třetí souřadnice .....	30
Obrázek 27: Optimalizace počtu zobrazených bodů u záznamů .....	31
Obrázek 28: Limit zobrazených záznamů .....	32
Obrázek 29: Vývojový diagram vykreslení a optimalizace.....	33
Obrázek 11: Struktura aplikace.....	34
Obrázek 30: Silverlight struktura.....	37
Obrázek 31: Horní panel testovací aplikace .....	38
Obrázek 32: Testovací aplikace.....	39
Obrázek 33: Panely na mapě .....	40
Obrázek 34: Nemapový projekt IKE.....	42

# 1. Úvod

## 1.1 Historie kartografie

Člověk měl již od pradávna potřebu značit si trasu pro usnadnění orientace v terénu. S postupem času ve válečných taženích nešlo ani tak o sílu a počet vojsk protivníka, ale o získání strategicky výhodného místa. Vůdci posílali průzkumníky pro zaznamenání míst a plánování taktiky bojů. Tyto jednoduché mapy pomáhaly při vymýšlení válečné strategie a rozmisťování vojsk. Tudíž člověk kladl nemalé požadavky na kvalitu zpracování map.

Zaznamenávání různých údajů do mapy pomohlo lidstvu např. při sledování průběhu epidemie, kdy zjištěná místa výskytu pomohla najít příčiny. Zajímavým příkladem může být vypuknutí cholery v roce 1854 na Broad Street (Londýn).[1] V polovině 19. století měla oblast Soho vážný problém s nedostatkem hygienických služeb. Mnohé domy měly sklepy se septiky v půdě pod podlahou. Jelikož byly žumpy přeplněné, rozhodla vláda v Londýně přemístit skládku odpadů do řeky Temže. Tato akce kontaminovala vodu, což vedlo k vypuknutí cholery. Do konce roku, v místě ohniska, zemřelo 616 lidí. John Snow – vyšetřovatel epidemie – zaznačil do mapy případy úmrtí na cholera. Zjistil, že většina úmrtí byla soustředěna kolem studny. Ukázal, že vodárenské společnosti braly vodu ze znečištěných částí Temže a dodávaly do domácností vodu se zvýšeným výskytem cholery. Zanesení výskytů úmrtí do mapy pomohlo určit místa působení a příčinu epidemie.



Obrázek 1: Mapa epidemie dle Johna Snow

Zdroj ilustrace: [http://en.wikipedia.org/wiki/1854\\_Broad\\_Street\\_cholera\\_outbreak](http://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak)

## 1.2 Kartografie v současnosti a vliv na člověka

Schopnost kombinovat data v geografickém kontextu je pro společnost a uživatele stále důležitější. Mapy mohou sloužit ke sledování povodní či zaznamenání ohrožených míst.

Díky předpovědi počasí zakreslené v mapě víme, jak se máme na další dny obléknout, popř. jak si naplánovat své aktivity.

S vývojem techniky se možnosti využití map v reálném světě stávají každodenní součástí života každého z nás. Při cestě do neznámých míst již nemusíme být vybaveni pouze papírovou mapou, která není schopna zaznamenat změny v čase a v průběhu let se stává značně neaktuální. Vznikají nové cesty, silnice, ulice a tam, kde kdysi vedla cesta, dnes stojí moderní budovy. V těchto případech se papírová mapa stává neužitečnou. Dříve armádou využívané GPS navigace se značně rozšířily do civilního segmentu. Kvalitní satelity rozeseté po obloze nám pomáhají určit polohu s přesností v řadech několika jednotek metrů. Rozšíření těchto zařízení je v dnešní době obrovské. GPS senzor již není vzácností ani v průměrném mobilním telefonu.

Určením naší aktuální polohy na mapě můžeme využít k zaznamenání dobré restaurace a díky internetu ihned informovat naše přátele či známé.

Lidé celosvětově hrají s pomocí GPS turisticko-sportovní hru zvanou geocaching. Základní myšlenkou je najít venku skryté nádoby, tzv. geocache, a pak se podělit o své zážitky na internetu.

Chytré využívání map může mít vliv na miliony lidí denně cestujících do práce. Díky včasnému zaznamenání autonehody nebo uzávěrky silnice může dobře navržený GPS systém upozornit řidiče vozu na tuto událost a ihned mu nabídnout alternativní cestu. Pouhá informace o právě vzniklé dopravní zácpě, může mít vliv na to, zda řidič přijde včas do práce a ušetří si i několik desítek minut a napomenutí šéfa.

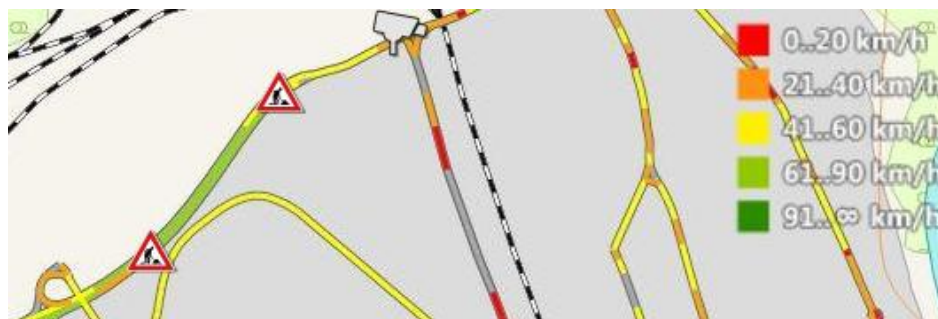
## 2. Elektronické mapy

Obyčejné mapy vycházejí z módy. Naproti tomu ty chytré jsou dnes snad všude. Stačí si představit plán města, na němž jsou vyznačeny jednosměrné silnice a uzavřené silnice. To se dá řešit buď hloupě, tedy že se s každou změnou překreslí celá mapa, nebo chytrě, že se na jednu průhlednou fólii nakreslí jednosměrné silnice, na druhou uzavřené silnice a obě se položí na mapu. Změny se pak dělají pouze na fóliích. Takové řešení, které někdy docela stačí, se dá považovat za primitivní GIS. Skutečný, tedy počítačový GIS spočívá v tom, že jednotlivé fólie (říká se jim vrstvy) mají datovou podobu a počítač na přání složí dohromady jakoukoli jejich kombinaci. Jednotlivé vrstvy mohou obsahovat i velmi rychle se měnící data (například o hustotě dopravy) a do výsledků se promítá vždy jejich aktuální podoba. Úžasné je, že díky standardům GIS nemusí být data, která tvoří jednotlivé vrstvy, uložena v počítači uživatele. Lze k nim přistupovat přes internet.[2]

Volně dostupných služeb nabízejících na internetu zobrazování informací na mapových podkladech je celá řada. Nevýhodou je nutnost opustit webový portál pro pohodlnou práci s datovými informacemi. Proto se nabízejí vývojářům technologie, které dovolí přizpůsobit si a následně implementovat zobrazení map do webových stránek. Mapy je možné vykreslit mnoha způsoby. Nejrozšířenější jsou mapy vykreslené pomocí Javascriptu. Takto vykreslené mapy nabízí např. Google, Microsoft (Bing maps), OpenStreetMap, Mapy.cz a další. Dále vznikají další projekty nabízející vykreslené mapy např. v prostředí Flash ([www.flashearth.com](http://www.flashearth.com)), ale nejsou příliš rozšířené. Se vzrůstající oblibou prostředí MS Silverlight[3] vznikají mapové aplikace i zde.

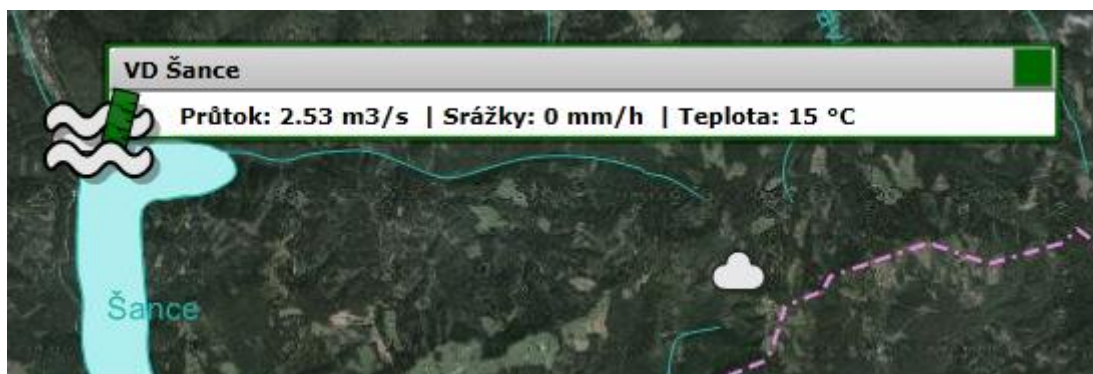
### 2.1 Příklady využití zobrazení dat na mapovém podkladu v reálných aplikacích

Zobrazování aktuálního stavu dopravní situace na silnicích (dopravní zácpy, nehody, údržba silnic...) umožňuje projekt Floreon+ (Obrázek 2). Tento projekt je zaměřen na vývoj integrovaného systému pro modelování, predikci, monitorování a podporu zvládání krizových situací pro potřeby Moravskoslezského kraje.[4] Floreon je dostupný na adrese <http://floreon.vsb.cz/web/Mapa.aspx?type=T&lang=cs>.



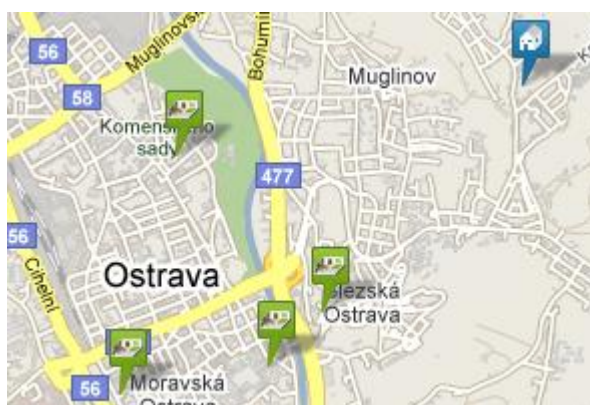
Obrázek 2: Zobrazení aktuálního stavu na silnicích v projektu Floreon

Další oblastí zaměření tohoto projektu je sledování aktuální meteorologické situace a situace na vodních tocích (Obrázek 3).



Obrázek 3: Sledování situace na vodních tocích v projektu Floreon

Hledání a zobrazování nemovitostí (k prodeji, pronájmu, zajímavostí poblíž nemovitostí) nabízí realitní server [www.mapanemovitosti.cz/](http://www.mapanemovitosti.cz/) (Obrázek 4).



Obrázek 4: Zobrazení nemovitostí na prodej na realitním serveru Mapa Nemovitostí

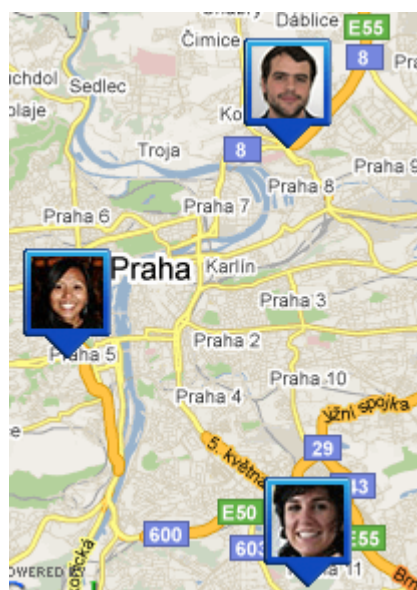
Vyvěšení zajímavých fotografií, virtuální prohlídka míst, která se uživatel chystá navštívit, náhled záznamů webových kamer v závislosti na kontextu – tyto a podobné služby nabízí webový portál Panoramio.com (Obrázek 5).





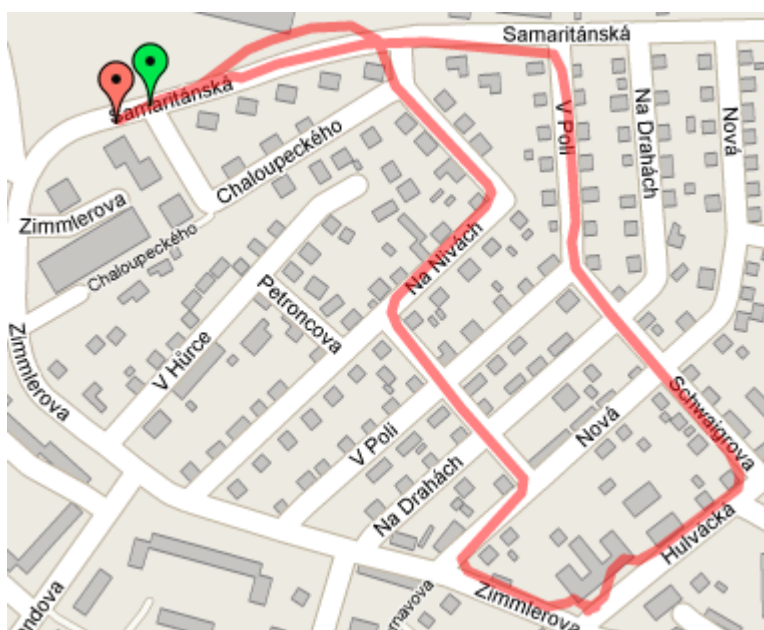
Obrázek 5: Fotografie zajímavých míst

Služba Google Latitude umožňuje sledování polohy přátel. Snadno se tak s nimi můžete sejit, když náhodou zjistíte, že jsou poblíž (Obrázek 6).



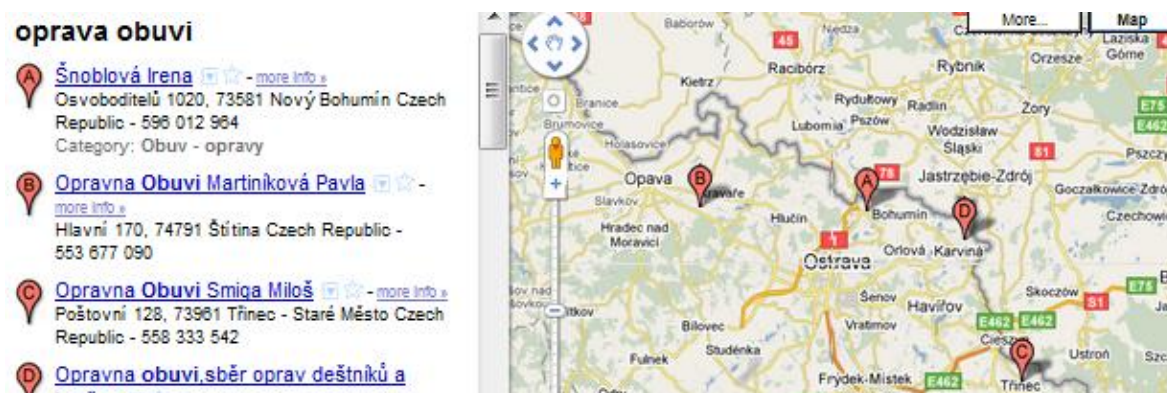
Obrázek 6: Sledování polohy přátel

Plánování a zaznamenávání tras při sportu (cyklistika, turistika, běh na lyžích...) lze jednoduše přes službu Google Maps. Trasu si můžete předem připravit v počítači přes <http://www.google.com/maps> a následně nahrát do své GPS navigace. Ta vás dle vytvořené trasy bude navádět. Další možností je zaznamenat si svou trasu při provádění aktivity (např. běh) navigací GPS a potom importovat získaná data do Google Maps. Jednoduše tak získáte celkový pohled nad vaší trasou (Obrázek 7).



Obrázek 7: Zaznamenaná trasa v Google Maps

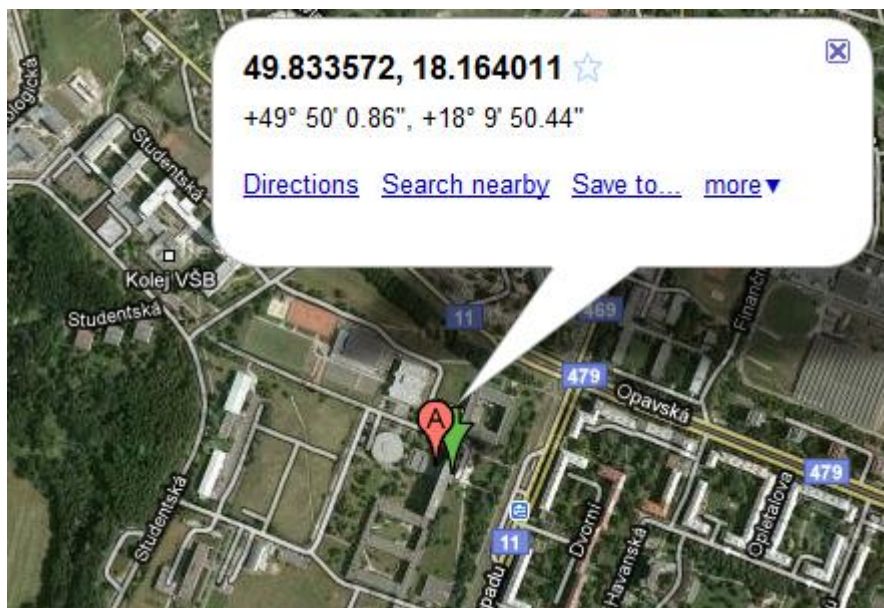
Vyhledávání nejbližších míst (kavárny, restaurace, hotely, supermarkety...), vyhledávání služeb (oprava obuvi, cykloservis, kadeřnictví...) a následné zobrazení rozmístění, z něhož lze vyčíst hluchá místa pro pokrytí trhu touto službou nabízí např. Google Maps (Obrázek 8).



Obrázek 8: Zobrazení míst, kde je možné si nechat opravit obuv

## 2.2 Souřadnicový systém elektronických map

Pro určení polohy bodu na mapě je nutné vycházet z jednotné soustavy souřadnic. Mapové aplikace využívají eliptický souřadnicový systém WGS84, který je celosvětově nejrozšířenější (Obrázek 9).



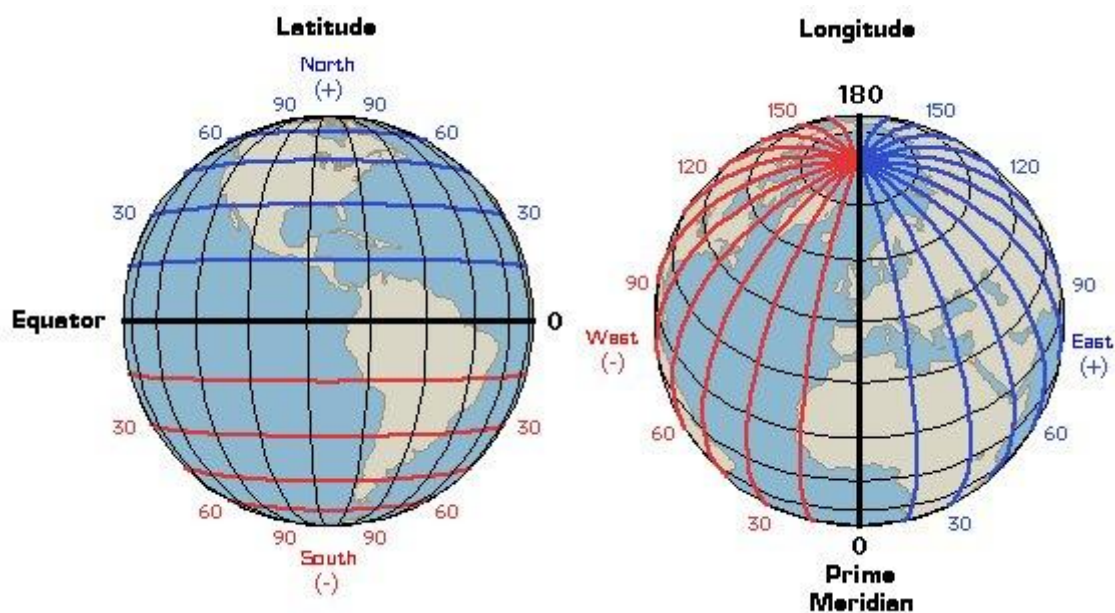
Obrázek 9: Souřadnice VŠB-TU Ostrava

Souřadnicový systém WGS84 je pravotočivá kartézská soustava souřadnic se středem v těžišti Země (včetně moří a atmosféry). Kladná osa x směřuje k průsečíku nultého poledníku a rovníku, kladná osa z k severnímu pólu a kladná osa y je na obě předchozí kolmá ve směru doleva ( $90^\circ$  východní délky a  $0^\circ$  šířky), tvoří tak pravotočivou soustavu souřadnic.[5]

Souřadnice je definovaná pomocí zeměpisné šířky (longitude) a zeměpisné délky (latitude). Pomocí těchto souřadnic můžeme určit libovolné místo na Zemi. Termín zeměpisná délka je definován jako horizontální linky jdoucí od východu na západ. Horizontální linky jsou značeny od nuly stupňů u rovníku až do  $90^\circ$  k severnímu pólu a  $90^\circ$  k jižnímu pólu (Obrázek 10 vlevo). Podélné linky jsou vzájemně paralelní.

Zeměpisná délka určuje polohu povrchu Země směrem k východu nebo západu od Greenwichského poledníku (Obrázek 10 vpravo).





Obrázek 10: Souřadnicové osy

Zdroj: <http://geographyworldonline.com/tutorial/latitudelongitude.jpg>

Je to úhel, který svírá rovina místního poledníku, procházejícího určeným bodem a rovina Greenwichského (nultého) poledníku. Body ležící na východ od nultého poledníku tj. na východní polokouli až do hodnoty  $180^\circ$  mají zeměpisnou délku východní (kladnou). Body ležící na západ od nultého poledníku tj. na západní polokouli až do hodnoty  $180^\circ$  mají zeměpisnou délku západní (zápornou). [6]

## 3. Vizualizační mapová komponenta pro prostředí MS Silverlight

Vývoj webové aplikace začíná specifikací požadavků. Jedná se o souhrn bodů popisujících základní vlastnosti kladené na budoucí systém. Tyto body budou postupně analyzovány a na základě analýzy a návrhu bude následně implementována webová aplikace splňující právě tyto požadavky.

Při vývoji testovací aplikace se musí brát zřetel na to, že později bude odděleno její „pevné jádro mapové komponenty“, aby vznikla samostatná komponenta, která bude snadno implementovatelná do dalších systémů v prostředí MS Silverlight.

### 3.1 Požadavky na testovací prostředí a mapovou komponentu

- Bude se jednat o aplikaci běžící v prostředí MS Silverlight.
- Bude obsahovat mapovou komponentu, která bude do testovací aplikace integrována. Proto je nutné zmapovat možnosti dostupných mapových komponent.
- Mapová komponenta by měla být volně dostupná a měla by pracovat bez dalších omezení.
- Pomocí grafického uživatelského rozhraní bude možné u této komponenty měnit mapový podklad. Mapová komponenta nesmí být závislá na jediném zdroji mapového podkladu. Zvolený mapový podklad by měl být snadno měnitelný za běhu aplikace.
- V rámci testování komponenty bude možné použít další komponenty třetích stran, které rozšíří možnosti GUI.
- Vytvořená testovací aplikace by měla prezentovat zobrazení informačních vrstev a datových formátů. Tyto mapové vrstvy budou do webové aplikace načteny z externích zdrojů zadaných dle URL.
- Testovací aplikace bude vyvíjena takovým způsobem, aby na jejím základě bylo možné vytvořit vlastní mapovou komponentu s uvedenou funkcionalitou.
- Ovládání aplikace by mělo být intuitivní – práce s mapou by měla být taková, na jakou jsou uživatelé zvyklí z jiných mapových aplikací, tzn. posun po mapě uchopením mapy a tažením myši, přiblížení a oddálení pomocí rolovacího kolečka myši, apod. (popř. pomocí klávesnice – posouvání šipkami, přiblížení a oddálení tlačítka „+“(plus) a „-“(mínus)).

#### 3.1.1 Požadavky na nastavení základní konfigurace

- Změnu mapového podkladu
- Zobrazení lupy
- Zobrazení měřítko
- Zobrazení navigace
- Zobrazení souřadnic mapy
- Výpis informačních zpráv na mapě

- Omezení oblasti mapy na ČR
- Nastavení souřadnic na mapě

### 3.1.2 Požadavky na práci s informačními vrstvami nad daty

- Načtení a zobrazení dat ze SHP souboru
- Načtení a zobrazení dat z XML souboru
- Možnost o budoucí přidání dalších formátů

## 3.2 Možnosti práce s mapou v prostředí MS Silverlight

Komponent nabízející mapové služby v prostředí MS Silverlight je celá řada. V této kapitole si je popíšeme a zvolíme si komponentu, která nejlépe splňuje naše požadavky.

### 3.2.1 BruTile – GIS tiling library

Jedná se o projekt pod GNU LGPL licenci, který vychází z mapové knihovny SharpMap. SharpMap je komponenta umožňující práci s mapou ve webových i desktopových aplikacích. Samotná SharpMap komponenta nenabízí podporu pro Silverlight aplikace, ale její tvůrci plánují začlenit tuto funkcionalitu do budoucích verzí. Do té doby může být vhodnou náhradou právě knihovna BruTile. Tato knihovna je volně ke stažení na <http://brutile.codeplex.com/>.

*BruTile umožňuje:*

- Vykreslení bodů, mnohoúhelníků, multibodů, multipolygonů, atd.
- Načtení vrstev z OpenStreetMap, BingMaps, Geodan's geoserver.nl a další
- Využívá WMS Tile Caching (cílem je optimalizace přenosu mapových podkladů přes Internet)
- Zoom a skládání mapových dlaždic (panning)
- Běží na platformě Silverlight verze 3.0

Nevýhodou tohoto projektu je nedostatečná dokumentace a nejistá budoucnost. Až SharpMap nabídne podporu pro prostředí Silverlight, tato komponenta nejspíš nebude mít již co nabídnout. Komponenta SharpMap nabízí velkou podporu rastrových a vektorových datových formátů (včetně ESRI ShapeFile), mapových vrstev a podrobnou dokumentaci.

### 3.2.2 Map Suite 3.0 Silverlight Edition

Map Suite je velice rozsáhlý komerční mapový produkt, který již v základu podporuje práci se ShapeFile soubory. Zvládá klasické operace s mapou, jako je např. přiblížení, posouvání mapy, ovládání rolovacím kolečkem, interakce s tvary na mapě, a další.

Obsahuje podporu pro prostorové formáty ESRI ShapeFile, Oracle Spatial, Microsoft SQL 2008, PostgreSQL/PostGIS, GRID, popř. umožňuje vytvořit nový OpenGis® zdroj dat. Umožňuje

dynamicky přidávat do mapy vlastní ikony, tvary, linie, body nebo dokonce grafy. Pro načtení podkladových vrstev a textur nabízí formáty dat jako např. JPEG2000, ECW, MrSid a GeoTIFF.

Pro podporu vývoje a rychlejší seznámení s mapovou komponentou obsahuje na svých stránkách ukázkové aplikace se zdrojovým kódem.

Tento profesionální GIS produkt je bohužel znevýhodněn cenou. I když nabízí vše potřebné a zároveň některé funkce navíc, cena začíná na \$3495.

### 3.2.3 ArcGIS API for Microsoft Silverlight

ArcGIS API for MS Silverlight nabízí zobrazení interaktivních map. Vyžaduje ke svému běhu ArcGIS Server, ze kterého získává výsledky – data k zobrazení. Tento server není nutné vlastnit, ale stačí zadat URL zdroje, ze kterého chceme čerpat data.

ArcGIS API for MS Silverlight jako většina mapových komponent umožňuje přidání a kreslení polygonů, posouvání po mapě, přibližování. Tato komponenta navíc nabízí služby pro volání dotazů na ArcGIS Server, hledání vrstev v mapě dle atributů, hledání geografické lokace dle zadané adresy či hledání nejkratší cesty.

Bohužel tato komponenta postrádá podporu mapových služeb z jiného než ESRI zdroje. Tento nedostatek se projeví hlavně, když chceme načíst jiné mapové podklady, např. stále oblíbenější OpenStreetMaps apod.

### 3.2.4 Virtual Earth Silverlight Map Control

Virtual Earth Silverlight Map Control v sobě kombinuje mapovací funkcionalitu s bohatým grafickým frameworkem s podporou vývoje pod .NET prostředím. To umožňuje vývojářům rozšířit funkcionalitu platformy Microsoft Virtual Earth<sup>1</sup>. Pro podporu vývoje obsahuje Virtual Earth Silverlight Map Control interaktivní SDK s příklady demonstrujícími funkcionalitu jako je vrstvení prvků na mapě, hladší dlaždicové přechody, lepší události, přizpůsobení chování a integrace s webovými službami (Web Services) k provádění geokódování (konverze zeměpisné šířky/délky). Informace obsažené v této kapitole se opírají o texty uvedené v literatuře [7].

#### *Virtual Earth Silverlight Map Control oproti konkurenci nabízí:*

- Zdokonalené poskytnutí/vykreslení (rendering) map – skládání (panning) a zoomování používá směšování (blending) pro přechod mezi dlaždicemi, což je významné vylepšení oproti mapovým podkladům vykresleným pomocí technologie AJAX.
- Kvalitní grafický rámec – podpora animací WPF a transformace grafických objektů a obrázků.
- Využití rozsáhlých technologií .NET – Silverlight knihovny jsou podmnožinou .NET frameworku ve verzi 3.5. K vytváření aplikací je možné využít Visual

---

<sup>1</sup> Nyní propagován pod názvem Bing Maps, ale pojem Microsoft Virtual Earth je více zažitý a využívá se ve většině diskusních fór a v internetových článcích, ze kterých jsem čerpal.

Studio. K vytváření uživatelského rozhraní je možné použít nástroj Expression Blend.

- Umožňuje vytvořit konzistentní prostředí ve všech moderních webových prohlížečích. Odpadá nutnost ladění jako je tomu u JavaScriptové technologie AJAX.
- Oproti JavaScriptovým mapovým aplikacím nabízí Virtual Earth v prostředí Silverlight vyšší výkon. Již při vykreslování pár set POI nebo poligonů s tisíci body naráží JavaScriptová aplikace na své limity a práce s ní již není tolik uživatelsky přívětivá. Virtual Earth Silverlight Map Control nabízí mnohem plynulejší chod aplikace i při vykreslení tisíců POI.
- Díky zasazení Virtual Earth platformy do prostředí Silverlight se nám nabízí vývoj mnohem bohatších mapových aplikací.

Kombinace technologie Silverlight a Virtual Earth nabízí možnost k vývoji inovativních nápadů, které snadno zachytí a udrží pozornost uživatele. Virtual Earth Silverlight Map Control dává vývojářům volnost při vytváření lepší verze Virtual Earth aplikací. Díky technologiím Silverlight a Virtual Earth je zoomování a posouvání mapy okamžité a bezproblémové. Umožňuje jednoduchou integraci animace a práci s dynamickým obsahem. Bohužel tato komponenta nepodporuje načtení dat z dalších mapových formátů, ale tento nedostatek se dá vyřešit naprogramováním. Pro snadnější seznámení s mapovou komponentou obsahuje Virtual Earth sadu volně dostupných příkladů i se zdrojovými kódy a podrobnou nápovědou což velmi usnadní následný vývoj.

Bez nutnosti práce nabízí Virtual Earth napojení na mapové podklady získané pomocí satelitních senzorů. Další mapové podklady je možné dynamicky připojit.

#### *Kroky k napojení vlastní datové vrstvy v prostředí VE Silverlight Map Control[8]:*

1. Vytvořit vlastní objekt Tile Source (dědí ze třídy TileSource)
  - a. Definovat „uriFormat“ pro získání Uri obrázku mapy
  - b. Přepsat „GetUri“ metodu. Tato metoda využívá „uriFormat“ za účelem vytvoření Uri obrázku mapy
2. Přidat Objekt Tile Source do mapy
  - a. Vytvořit instanci objektu MapTileLayer
  - b. Vytvořit instanci našeho vlastního objektu Tile Source
  - c. Přidat vlastní Tile Source objekt do MapTileLayer
  - d. Nastavit průhlednost MapTileLayer na požadovanou hodnotu
  - e. Přidat MapTileLayer jako potomka do kolekce mapy

Metoda „GetUri“ objektu Tile Source obsahuje následující tři parametry:

Souřadnice X – zeměpisná délka

Souřadnice Y – zeměpisná šířka

zoomLevel – úroveň přiblížení mapové dlaždice

Prakticky to znamená, že programátor si zvolí, jaký mapový podklad bude chtít v komponentě zobrazit. Dle zvoleného podkladu se odešle http požadavek na server s volně dostupnými mapovými podklady.

Např. <http://tile.openstreetmap.org/{2}/{0}/{1}.png>, kde za parametr {2} se dosadí aktuální zoom, za parametr {0} souřadnice X (zeměpisná délka), za parametr {1} souřadnice Y (zeměpisná šířka). Vykreslení a poskládání mapového podkladu z takto stažených dlaždic již obstará Virtual Earth for Silverlight.

Obdobným způsobem je možné připojit do aplikace další mapové podklady. V některých případech je nutné konvertovat souřadnice X, Y, a zoomLevel do správného tvaru „uriFormat“ k načtení obrázků mapových dlaždic.

Jelikož se jedná o produkt z dílny Microsoft, budoucí vývoj a následná podpora je zaručena. Používání Virtual Earth je pro studijní a nekomerční projekty zdarma. Po naprogramování podpory dalších datových formátů bude tato komponenta splňovat všechny naše požadavky, a proto se budeme zabývat vývojem aplikace právě s touto mapovou komponentou.

## 4. Typy geografických dat zobrazených na mapě

Geografická data mohou být na mapové komponentě prezentovány dvěma způsoby:

- rastrově
- vektorově

Základem rastrového modelu je buňka, která nese libovolnou hodnotu. Uložení dat je provedeno pomocí imaginární sítě – rastru, ve kterém je každá buňka stejně velká. Dvoudimenzionální rastrová buňka se nazývá „pixel“. Tento datový model je vhodný pro uložení obrazových dat – např. naskenovaná mapa.

Reprezentace geometrie vektorových dat je tvořena vektory nebo posloupnostmi vektorů. Pomocí vektorového modelu můžeme na mapě zobrazit bod zájmu, mnohoúhelník nebo lomenou čáru.

Výhody vektorové grafiky:

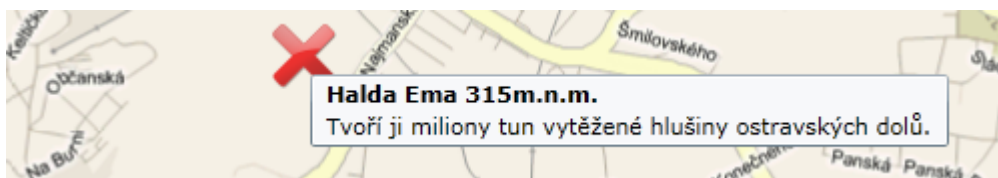
- je možné zmenšovat nebo zvětšovat geoprvek bez ztráty kvality
- je možné pracovat s každým geoprvkem v obrázku odděleně
- výsledná paměťová náročnost obrázku je obvykle mnohem menší než u rastrové grafiky
- vysoká polohová přesnost a relativně malý objem uložených údajů

Proto je výhodné při práci s geografickými daty využít vektorového modelu.

### 4.1 Bod zájmu (point of interest)

Pomocí bodu zájmu můžeme zvýraznit určité místo na mapě (

Obrázek 11). Definujeme jej souřadnicemi v prostoru. Toto místo může obsahovat další volitelnou atributovou složku, která je vyznačena na mapě pomocí ikony, popř. textového popisu. V praktické závislosti na kontextu se může jednat např. o informaci o uzavřené silnici, opravě mostu, místo s častým výskytem nehod v souvislosti s dopravou nebo označení restaurace, hotelu, aquaparku v souvislosti s rekreací. Občas se můžeme setkat v souvislosti s mapovými aplikacemi s termínem značka (placemark) nebo připínáček (pushpin).



Obrázek 11: Zobrazení bodu zájmu na mapovém podkladu



## 4.2 Lomená čára (polyline)

Lomenou čarou můžeme zobrazit na mapě cesty, koleje apod. Obrázek 12 znázorňuje kolejovou trať vykreslenou pomocí objektu typu lomená čára. Jedná se o souhrn souřadnic propojených úsečkou.



Obrázek 12: Zobrazení lomené čáry na mapovém podkladu

## 4.3 Mnohoúhelník (polygon)

Vymezuje mapovou oblast pomocí mnohoúhelníku. Definujeme jej jako uzavřenou linii zvýrazněnou barvou výplně. Může znázorňovat různé typy rozsáhlých objektů – např. zalesněná oblast, okres, vodní plochy apod. Obrázek 13 znázorňuje vykreslenou vodní plochu pomocí mnohoúhelníků.



Obrázek 13: Zobrazení mnohoúhelníku na mapovém podkladu



## 5. Ukládání geografických dat

Abychom mohli vektorová data importovat do mapy, je nutné znát strukturu souborů, ve kterých jsou data uložena. K uložení geografických dat slouží různé datové soubory.

Nejběžnějším datovým souborem pro přenos nejen vektorových dat je soubor typu KML. KML soubory jsou vhodné pro ukládání bodů zájmu. Mohou být velice variabilní – jejich struktura a nesené informace se dá lehce změnit a přizpůsobit požadavkům programátora. Jedná se o textový soubor snadno čitelný i v textovém editoru (plain text). Snadno se exportuje a importuje do jiných aplikací.

Soubory typu ShapeFile jsou přizpůsobeny pro uložení většího počtu dat. Na internetu se dají volně stáhnout ShapeFile soubory obsahující geografické vrstvy jako např. kraje, vodstvo, železniční tahy, silnice apod. Tyto soubory mohou být využity v mapových aplikacích pro vizualizační nebo analytické úlohy. ShapeFile je uložen v binárním tvaru, což jej činí nečitelný v běžném textovém editoru, ale vyniká rychlejší práci v rámci souborového systému.

### 5.1 Soubor typu ShapeFile

Jedná se o prostorový datový formát vyvinutý Environmental Systems Research Institute, Inc. (ESRI).

ShapeFile slouží k ukládání geometrických informací i s atributy pro vektorové i prostorové funkce v souboru dat. Podporuje ukládání bodu (point), linie (line) a plochy (area). Plochy jsou reprezentovány jako uzavřené smyčky – dvourozměrné polygony. Atributy jsou uloženy v souboru typu dBASE®. Každý atribut záznamu má vztah jedna ku jedné k příslušnému záznamu (record).

*ShapeFile může být vytvořen následujícími metodami[9]:*

- Exportováním – ShapeFile může být vytvořen exportováním datového zdroje použitím ARC/INFO®, PC ARC/INFO®, Spatial Database Engine™, ArcView® GIS nebo BusinessMAP™.
- Digitalizací – ShapeFile může být vytvořen přímo digitalizací tvarů pomocí nástroje ArcView® GIS.
- Programováním – použitím Avenue™ (ArcView® GIS), MapObjects™, ARC Macro Language (AML™)(ARC/INFO) nebo Simple Macro Language (SML™) (PC ARC/INFO®)
- Přímým zápisem ShapeFile specifikace vytvořená programem.

ShapeFile vyvinutý firmou ESRI se skládá z hlavního souboru, indexového souboru a dBASE tabulky. Hlavní soubor je přímo přístupný soubor s variabilní délkou záznamu, ve kterém každý záznam popisuje tvar (shape) seznamem jeho vrcholů. V indexovém souboru každý záznam obsahuje pozici/odchylku (offset) k příslušnému záznamu z hlavního souboru od začátku hlavního souboru (tzn. jedná se o odkaz na začátek záznamu v hlavním souboru). Tabulka dBASE obsahuje

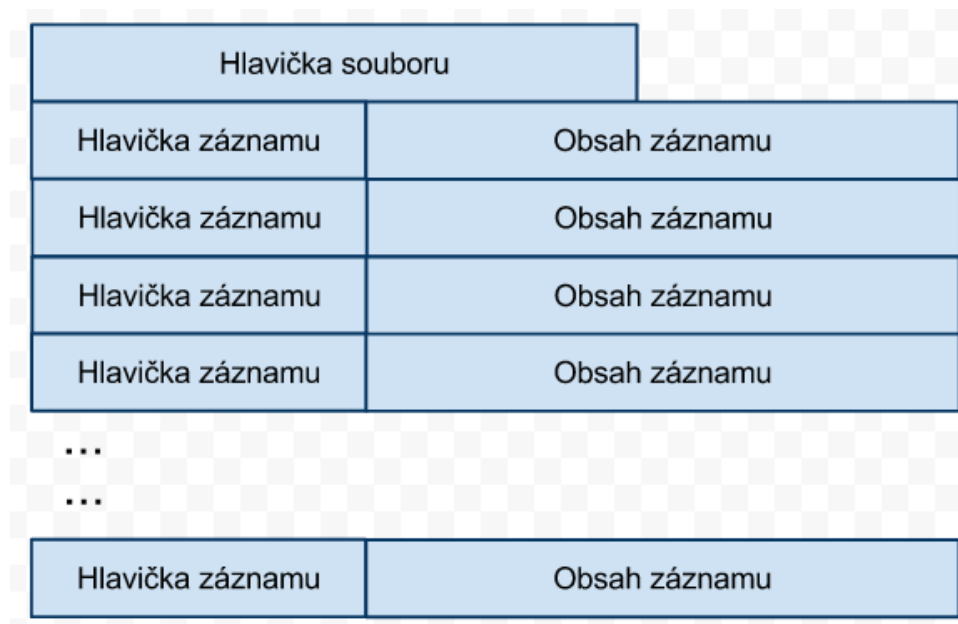
popisné atributy k jednotlivým záznamům. Vztah jedna ku jedné mezi geometrickým tvarem a atributem je založen na čísle záznamu. Atributy z dBASE jsou ve stejném pořadí, v jakém jsou záznamy v hlavním souboru.

Konvence pro pojmenování ShapeFile souborů[9]:

- Všechny soubory jsou pojmenovány podle 8.3 notace. (8 znaků slouží pro název, 3 znaky pro příponu).
- Hlavní soubor, indexový soubor a dBASE soubor mají stejný název (odlišují se pouze příponou).
- Název začíná alfanumerickými znaky (a-Z, 0-9) a následuje až sedm znaků (a-Z, 0-9,\_,-)
- Přípona hlavního souboru je „.shp“
- Přípona indexového souboru je „.shx“
- Přípona dBase tabulky je „.dbf“
- Názvy jsou psány malými písmeny v case-sensitivních OS

Struktura hlavního souboru (SHP)

Hlavní soubor obsahuje hlavičku o pevné délce a je následován proměnnou délkou záznamů. Každý záznam s proměnnou délkou se skládá z hlavičky pevné délky a proměnné délky obsahu záznamu (Obrázek 14).



Obrázek 14: Struktura hlavního souboru (SHP)

Hlavička hlavního souboru je délky 100 byte a skládá se ze 17 polí. Obsahuje informace o délce souboru, verze, typu tvaru, min hodnota X, min hodnota Y, max hodnota X, max hodnota Y, min hodnota Z, max hodnota Z, min hodnota M, max hodnota M a další vyhrazená místa pro

vlastnosti. Hodnota „Z“ se vyplňuje u trojrozměrného zobrazení, hodnota „M“ je určena k uživatelem definovanému měření, které náleží danému bodu.

Typ tvaru (shape type) je dán číselnou hodnotou (Tabulka 1)[9]<sup>2</sup>.

Hodnota	Typ tvaru
<b>0</b>	Prázdný tvar (Null Shape)
<b>1</b>	Bod (Point)
<b>3</b>	Lomená čára (PolyLine)
<b>5</b>	Mnohoúhelník (Polygon)
<b>8</b>	MultiBod (MultiPoint)

Tabulka 1: nejzákladnější typy tvaru

V hlavičce každého záznamu je uložena informace o pořadí záznamu a délce obsahu záznamu. Pro hlavičku záznamu je vyhrazena délka 8 bytu.

Data uložená v ShapeFile záznamu závisí na typu tvaru. Délka záznamu závisí na počtu částí a vrcholů.

*ShapeFile má limity na které je třeba brát zřetel[10]:*

- Neumožňuje ukládání topologických informací – není uložen vztah mezi jednotlivými geometrickými objekty. Při ukládání se využívá „špagetový model“, který patří mezi nejjednodušší. Pro uložení sousedních polygonů může vzniknout redundance uložených souřadnic, protože společná linie je ukládána dvakrát, pro každý polygon zvlášť. Pro většinu prostorových analýz je tento model nevhodný, ale v jednodušších systémech pro pouhé zobrazení dat je díky jednoduchosti výkonný.[11]
- Hrany křivky nebo polygonů jsou definovány pomocí bodů. Rozteč bodů implicitně určuje rozsah, pro který jsou údaje užitečné. Vyšší měřítko zobrazuje výsledky jako zubaté prvky. Bylo by třeba použít dodatečné body k dosažení hladkého tvaru křivky. Pro lepší vykreslení hladké křivky potřebuje polygon více prostoru pro data než např. spline, která může zachytit plynulost různých tvarů efektivně. Žádný z ShapeFile typů nepodporuje ukládání aproximační křivky spline.
- Maximální velikost .shp a .dbf souborů nesmí překročit 2GB ( $2^{31}$  bitů), to znamená, že může pojmout okolo 70 milionu bodů(souřadnic).
- Specifikace[9] uvádí, že všechny tvary uložené v ShapeFile musí být stejného typu. Kvůli tomuto omezení je nutné studnu (jako bod), řeku (jako lomenou čáru) a jezero (jako polygon) uložit do tří oddělených souborů.

<sup>2</sup> Tabulka obsahuje nejzákladnější tvary. Ve specifikaci ESRI Shapefile Technical Description[2] jich je celkem 14

### Postup přidání ShapeFile do seznamu vrstev

1. Systém zobrazí okno pro zadání informací o vrstvě (název, typ, URL, cesta k ikoně)
2. Uživatel vyplní údaje a potvrdí
3. Systém přidá záznam do seznamu vrstev
4. Systém uloží seznam vrstev do IsolatedStorage

### Přidání geografických dat uložených v souboru ShapeFile do aplikace:

1. Zadání URL adresy k SHP
2. Aplikace načte stream dat pomocí shapeFileReader
3. Aplikace uloží všechny body do nové vrstvy (třídy ShapeFile)
4. Aplikace vykreslí všechny vrstvy (z třídy typu ShapeFile)

## 5.2 Soubor typu KML

KML je založen na XML jazyce a je zaměřený na ukládání informací pro geografické vizualizace. Geografická vizualizace zahrnuje nejen prezentaci grafických dat na Zemi, ale také vizualizaci navigační, tj. zobrazení ve smyslu kde jít a kde se podívat. Z tohoto pohledu KML je doplňkem k většině z klíčových stávajících OGC standardů včetně GML (Geography Markup Language), WFS (Web Feature Service) a WMS (Web Map Service). V současné době, KML 2.2 využívá některých geometrických prvků získaných z GML 2.1.2. Mezi tyto prvky patří bod, řetězec úseček, lineární kruh a polygon.[12]

Struktura XML dokumentu pro přidání datových vrstev částečně vychází z KML specifikace. Struktura souboru KML je pro naše potřeby příliš rozsáhlá. Proto nám budou stačit pouze základní atributy, které je možné v budoucnu dále rozšiřovat.

Může obsahovat dva typy uložených dat. Mohou to být body zájmů nebo záznamy. Příklad XML dokumentu s dvěma body zájmu (placemark) je uveden v Zdrojový kód 1. Příklad XML dokumentu se záznamem popisující mnohoúhelník je uveden v Zdrojový kód 2.

```
<?xml version="1.0" encoding="utf-8"?>

<ShapeFileXML>
  <ID>5</ID>
  <Name>jmeno</Name>
  <ShapeType>1</ShapeType>
  <Visible>true</Visible>
  <XMin>48.8</XMin>
  <XMax>49.8</XMax>
  <YMin>11.1</YMin>
  <YMax>12.1</YMax>

  <Points>
    <ShapePoint>
      <ID>1</ID>
      <Location>49.8, 11.1</Location>
      <URL>/Images/Icons/223.png</URL>
```

```

<Title>Title message</Title>
<Text>Long text message</Text>
</ShapePoint>

<ShapePoint>
  <ID>3</ID>
  <Location>48.8,11.1</Location>
  <URL></URL>
  <Title></Title>
  <Text>This is text in the bubble</Text>
</ShapePoint>
</Points>

</ShapeFileXML>

```

Zdrojový kód 1: XML dokument s dvěma body zájmu

Tyto typy souborů, které následně poslouží k implementaci datových vrstev do vlastní komponenty, se vždy skládají z hlavičky a samotné definice bodů nebo záznamů.

### Hlavička XML

Hlavička je definovaná tagy **ID**, **Name**, **ShapeType**, **Visible**, **XMin**, **Xmax**, **YMin**, **YMax**. Hodnoty těchto tagů určují, jakým způsobem mají být data XML souboru prezentována.

Hodnota tagu **ID** identifikuje XML soubor. Tento parametr musí být v rámci načtených vrstev jedinečný.

Oproti tomu tag **Name** je dobrovolný a slouží pouze potřebám programátora.

Soubory XML mohou obsahovat vždy jen jeden typ záznamu, stejně jako je tomu u souborů typu ShapeFile. Tento typ je definován v hlavičce XML souboru atributem **ShapeType**. Může nabývat již dříve uvedených hodnot (viz. Tabulka 1).

Hodnota tagu **Visible** ovlivňuje, zda má být při prvním načtení zobrazena vrstva. Může nabývat hodnot „TRUE“ a „FALSE“.

Rozsah načtených vrstev je dán hodnotami tagů:

- **XMin** – minimální hodnota souřadnice X (longitude)
- **XMax** – maximální hodnota souřadnice X (longitude)
- **YMin** – minimální hodnota souřadnice Y (latitude)
- **YMax** – maximální hodnota souřadnice Y (latitude)

### Definice bodů

Body slouží k uložení polohy a popisu dané souřadnice. Bodů může být v jednom XML souboru libovolné množství. Body mohou být znázorněny na mapě jako obrázek s popiskem (Obrázek 15) nebo textová bublina (Obrázek 16). XML dokument obsahující jeden nebo více bodů obsahuje v hlavičce u tagu **ShapeType** hodnotu „1“.

Tělo XML dokumentu je ohraničeno tagem **Points**, který může obsahovat více objektů **ShapePoint**. Jeden objekt **ShapePoint** nese informace pro přidání bodu zájmu, značek, popisků...

Pro přidání obrázku s popiskem (tooltip, Obrázek 15) obsahuje tag **ShapePoint** elementy **ID**, **Location**, **URL**, **Title**, **Text**, kde:

- **ID** – identifikuje bod zájmu (musí to být jedinečné číslo)
- **Location** – obsahuje souřadnice umístění značky
- **URL** – udává adresu obrázku daného bodu (např. při umístění obrázku do ClientBin/Images/Icons/ stačí zadat „Images/Icons/223.png“ )
- **Title** – obsahuje nadpis popisku
- **Text** – obsahuje text popisku



Obrázek 15: Obrázek s popiskem

Pro přidání bubliny s textem (Obrázek 16) obsahuje tag **ShapePoint** elementy **ID**, **Location**, **URL**, **Title**, **Text**, kde:

- **ID** – identifikuje bod zájmu (musí to být jedinečné číslo)
- **Location** – obsahuje souřadnice umístění značky
- **URL** – musí být prázdné
- **Title** – musí být prázdné
- **Text** – obsahuje text bubliny



Obrázek 16: Bublina s textem

### Definice záznamů

Záznamy slouží k uložení složitějších struktur jako je např. lomená čára nebo mnohoúhelník. Struktura XML dokumentu obsahujícího mnohoúhelník je uvedena viz Zdrojový kód 2 a vykreslena na Obrázek 17.

```

<?xml version="1.0" encoding="utf-8"?>
<ShapeFileXML>
  <ID>5</ID>
  <Name>jmeno</Name>
  <ShapeType>5</ShapeType>
  <Visible>true</Visible>
  <XMin>48.2302141006791</XMin>
  <XMax>51.122538186594</XMax>
  <YMin>11.7722354209113</YMin>
  <YMax>19.0506778037238</YMax>

  <Records>
    <ShapeRecord>
      <ID>1</ID>
      <NumberOfPoints>8</NumberOfPoints>
      <XMin>48.2302141006791</XMin>
      <XMax>51.122538186594</XMax>
      <YMin>11.7722354209113</YMin>
      <YMax>19.0506778037238</YMax>
      <FillColor>Blue</FillColor>
      <StrokeColor>Yellow</StrokeColor>
      <StrokeThickness>2</StrokeThickness>
      <Opacity>0.5</Opacity>
      <Locations>
        50.50474296311,15.3592715537238,0
        51.0431701976878,19.0287051474738,0
        49.4628200629473,17.8751406943488,0
        48.2704496062864,19.0506778037238,0
        48.9135264680933,15.3647647177863,0
        48.2302141006791,11.7722354209113,0
        49.4949416381553,13.4696231162238,0
        51.122538186594,12.1512637412238,0
      </Locations>
    </ShapeRecord>
  </Records>
</ShapeFileXML>

```

Zdrojový kód 2: XML dokument obsahující mnohoúhelník

Hlavička XML dokumentu je téměř stejná, jako byla popsána výše, ale tag **ShapeType** obsahuje hodnotu „3“ pokud se jedná o lomenou čáru anebo hodnotu „5“ pokud mají být souřadnice záznamu prezentovány jako mnohoúhelník (hodnoty jsou definovány v Tabulka 1: nejzákladnější typy tvaru. Tělo XML dokumentu se liší nahrazením objektu **ShapePoint** objektem **Records**. Ten obsahuje libovolný počet elementů **ShapeRecord**.

Pro přidání mnohoúhelníku či lomené čáry obsahuje **ShapeRecord** elementy **ID**, **NumberOfPoints**, **XMin**, **XMax**, **YMin**, **YMax**, **FillColor**, **StrokeColor**, **StrokeThickness**, **Opacity**, **Locations**, kde:

**ID** – identifikuje přidáný polygon (musí to být jedinečné číslo)

**NumberOfPoints** – počet souřadnic v **Records-Locations**

**XMin**, **XMax**, **YMin**, **YMax** – definují rozsah přidávaného polygonu

**FillColor** – barva výplně polygonu

**StrokeColor** – barva ohraničení



**StrokeThickness** – šířka ohraničení

**Opacity** – sytost barvy výplně (0 – průhledné, 1 – plná sytost)

**Locations** – souřadnice všech bodů polygonu



Obrázek 17: Mnohoúhelník

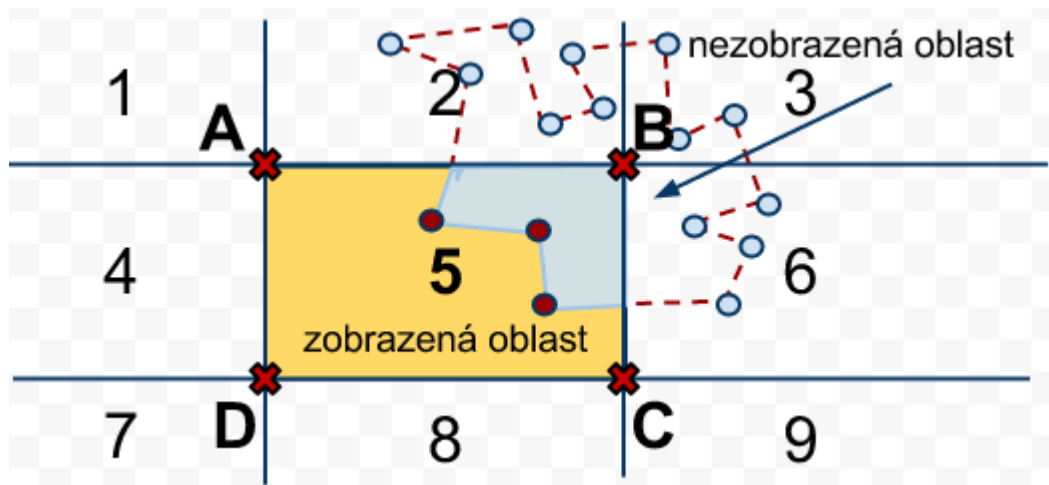


## 6. Optimalizace zobrazených dat

### 6.1 Rozdělení obrazovky na regiony

Už víme, jak dostat na mapový podklad vlastní vrstvy - geografická data načtená z externích souborů. Analýza nastavení mapové komponenty však ještě není u konce. Při bližším testování zjistíme, že práce s mapovou komponentou není vždy tak plynulá, jak by mohla být. Rozsáhlé soubory shapeFile tvoří vrstvy na mapě. Vrstvy tvořené tisíci body, liniemi a mnohoúhelníky jsou uloženy v paměti počítače. Mapová komponenta s nimi pracuje a nezáleží, jestli je vidíme všechny nebo zda všechny vrcholy leží mimo zobrazovanou oblast obrazovky (všechny souřadnice vrcholů jsou mimo aktuálně vykreslenou oblast mapy).

Z tohoto důvodu je vhodné nějakým způsobem rozdělit nezobrazovanou část obrazovky od zobrazované a pouze na zobrazované části vykreslovat vrcholy bodů. Tím se v některých případech výrazně urychlí práce s mapou a překreslování zobrazených vrcholů bude plynulejší. Nabízí se nám rozdělení obrazovky podle souřadnic rohových bodů zobrazené části. Tyto čtyři souřadnice jsou vyznačeny písmeny A-B-C-D (Obrázek 18). Protnutím bodů AB, BC, CD, DA, získáme čtyři přímky, které pomyslně rozdělí mapový podklad na devět částí. Prostřední část neboli také pátý region je jediná viditelná oblast obrazovky. Souřadnice ohraničující mnohoúhelník v tomto regionu má smysl vykreslit a dále s nimi pracovat. Souřadnice, které mají polohu mimo pátý region, je zbytečné ukládat do mapové komponenty, jelikož je uživatel momentálně nemůže vidět a zbytečně by tak zabíraly paměť a zpomalily práci s mapou.



Obrázek 18: Rozdělení obrazovky na regiony

Jsou dány souřadnice pro body A,B,C,D,X, kde:

- A = souřadnice levý horní roh
- B = souřadnice pravý horní roh
- C = souřadnice pravý dolní roh

D = souřadnice levý dolní roh  
X = souřadnice pro test regionu

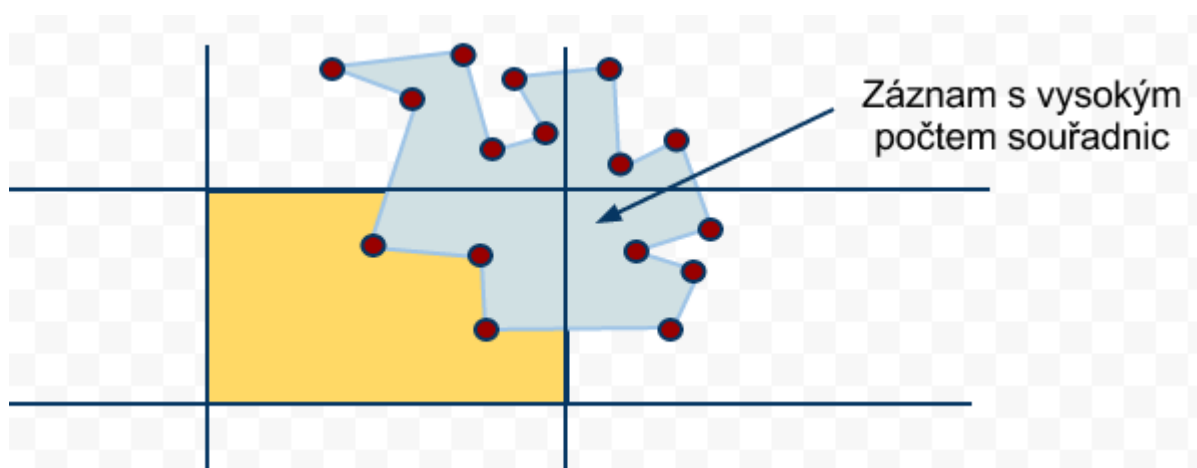
a potom regiony jsou určeny splněním těchto podmínek:

```
1 = (X.Latitude > A.Latitude) ^ (X.Longitude < A.Longitude)
2 = (X.Latitude > A.Latitude) ^ (A.Longitude < X.Longitude) ^
(X.Longitude < B.Longitude)
3 = (B.Latitude < X.Latitude) ^ (B.Longitude < X.Longitude)
4 = (X.Longitude < A.Longitude) ^ (A.Latitude > X.Latitude) ^
(X.Latitude > D.Latitude)
5 = (A.Latitude > X.Latitude) ^ (A.Longitude < X.Longitude) ^
(X.Latitude > C.Latitude) ^ (X.Longitude < C.Longitude)
6 = (B.Longitude < X.Longitude) ^ (B.Latitude > X.Latitude) ^
(X.Latitude > C.Latitude)
7 = (D.Latitude > X.Latitude) ^ (X.Longitude < D.Longitude)
8 = (D.Latitude > X.Latitude) ^ (D.Longitude < X.Longitude) ^
(X.Longitude < C.Longitude)
9 = (C.Latitude > X.Latitude) ^ (C.Longitude < X.Longitude)
```

kde číslo z intervalu 1..9 označuje pozici regionu(Obrázek 18) .

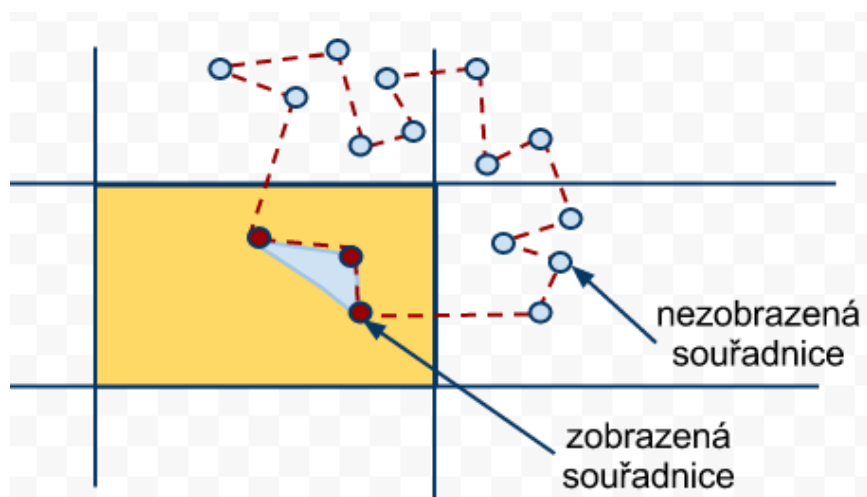
## 6.2 Optimalizace rychlosti práce s větším počtem bodů záznamů (dat)

U záznamů s velkým počtem souřadnic není výhodné přidat do mapové komponenty k vykreslení všechny souřadnice, protože by to velmi zpomalilo práci s mapovou komponentou. Při vyšším přiblížení nejsou v zobrazované oblasti obsaženy všechny souřadnice, proto je neefektivní přidávat do mapy souřadnice mimo aktuální obrazovku. (Obrázek 19)



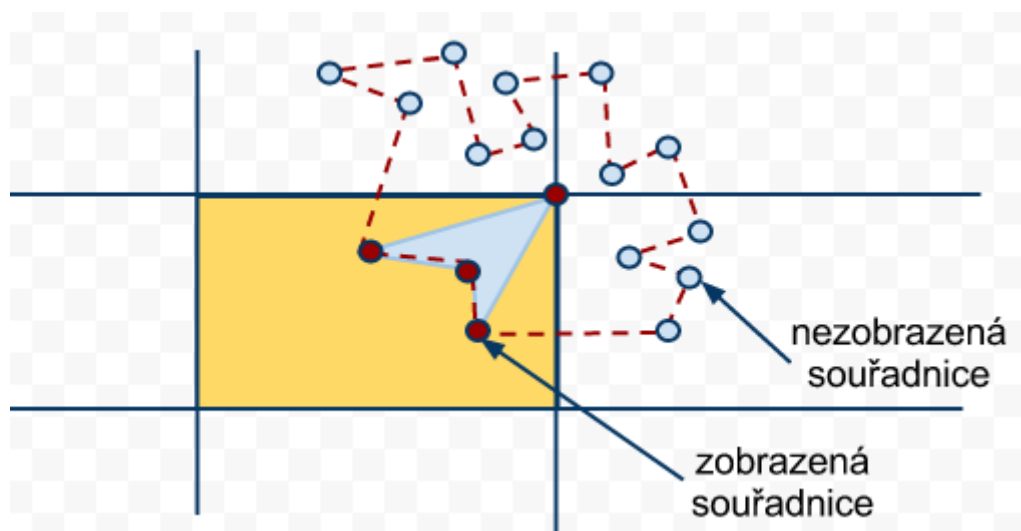
Obrázek 19: Vykreslení záznamu s vysokým počtem souřadnic

Ze zadaných souřadnic tedy vybere souřadnice ležící v 5. regionu (obdélník vymezující zobrazenou oblast obrazovky), ale ani toto není nejvhodnější řešení, i když se práce s mapou výrazně zrychlí. Počet bodů načtených do mapy se může výrazně snížit, ale v určitých případech se vrstva nemusí vykreslit správně. (Obrázek 20)



Obrázek 20: Nevhodně vykreslený mnohoúhelník

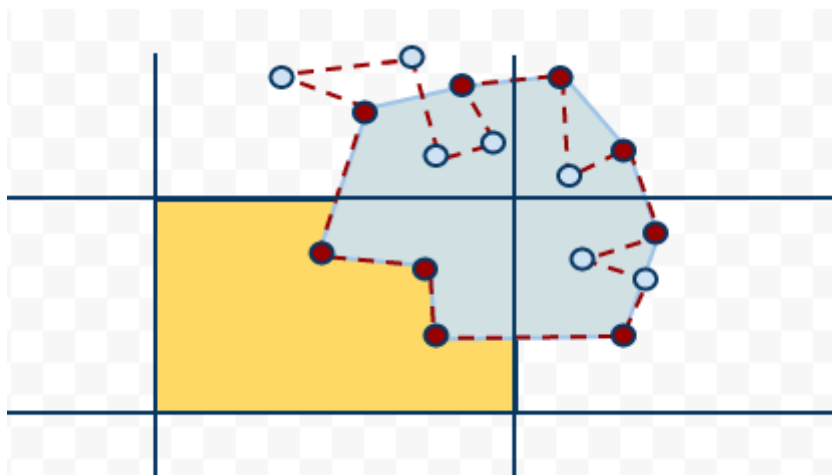
Tento problém vzniká, pokud záznam obsahuje body mimo zobrazenou oblast. Jedním řešením by bylo přidávat v místech, kde souřadnice přechází ze zobrazené oblasti do nezobrazené souřadnice A/B/C/D dle regionu ze kterého přechází a regionu, přes který se vrací zpět do zobrazované oblasti. Toto řešení má tu výhodu, že vykreslení už působí lepším dojmem a je použito minimum bodů, ale má to nevýhodu při zobrazení záznamu s větším rozestupem bodů. V tomto případě přechody se vykreslí zkresleně. (Obrázek 21)



Obrázek 21: Špatně vykreslený mnohoúhelník po přidání pomocných souřadnic

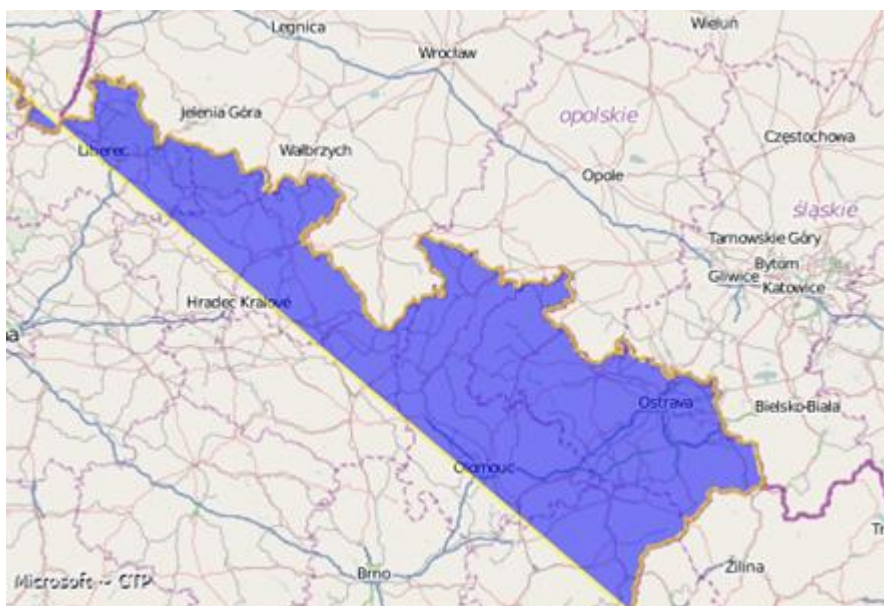
Další možností je přidat souřadnice, které následují po změně z jednoho regionu do druhého. Tímto způsobem vykreslený záznam působí kompletním dojmem, není narušený přechod

ze zobrazené oblasti do nezobrazené (Obrázek 22). Je použito pouze pár souřadnic, které nejsou vidět, takže rychlost s mapou není nijak zvlášť ovlivněna. Tato metoda se nazývá minimální ohraničující obdélník (minimum bounding rectangle), zkráceně bounding.

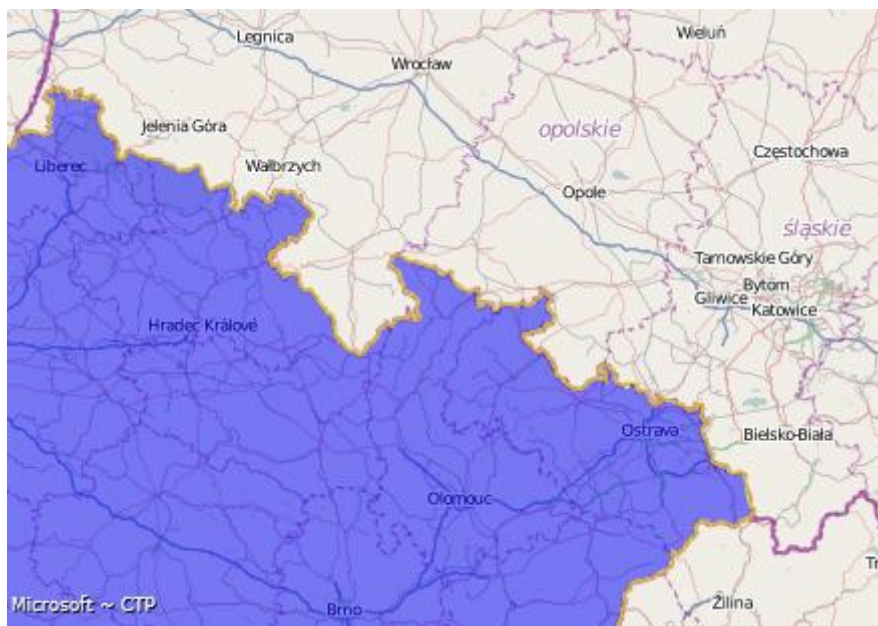


Obrázek 22: Správně vykreslený mnohoúhelník

Příklad špatně a správně vykreslené mapy přímo v prototypu aplikace (Obrázek 23 a Obrázek 24):



Obrázek 23: Špatně vykreslený mnohoúhelník obrysu ČR



Obrázek 24: Správně vykreslený mnohoúhelník obrysu ČR

Nyní je vykreslení záznamu je korektní, ale další problém s rychlosti může nastat, pokud daný záznam obsahuje hodně souřadnic i v zobrazené oblasti. Tento nedostatek je řešen pomocí zobrazení každé  $n$ -té souřadnice.

Výpočet se odvíjí od počtu souřadnic vrcholů a počtu záznamů umístěných v zobrazené oblasti. Postup této optimalizace je popsán v následující kapitole.

### 6.2.1 Proměnné ovlivňující způsob vykreslení

Než přikročíme k popisu optimalizačních algoritmů a podmínek pro jejich užití, je zapotřebí osvětlit některé pojmy (proměnné), které při tom budeme používat.

#### MALO

Označuje, kolik vrcholů má být přidáno do mapové aplikace. Jedná se o limit počtu souřadnic, které budou později zobrazeny. Nevhodně zvolená hodnota MALO může mít vliv na rychlost práce s mapou (příliš vysoká hodnota na pomalém počítači) nebo příliš zkreslené vykreslení mnohoúhelníku (nízká hodnota).

#### MINIMALNI ZAZNAM

Definuje minimální počet vrcholů, který má mít vykreslený záznam. Záznam s nižším počtem vrcholů bude mít vykresleny všechny jeho vrcholy. Záznam s vyšším počtem vrcholů bude optimalizován právě na tuto hodnotu (viz bounding).

### MALO\_ZAZNAMU

Jedná se o dolní hranici počtu záznamů, kdy má být použita optimalizace pomocí zobrazení každé N-té souřadnice vrcholu.

### HODNE\_ZAZNAMU

Jedná se o horní hranici počtu záznamů, kdy má být použita optimalizace pomocí zobrazení každé N-té souřadnice vrcholu.

### PCT\_ZAZNAMU

Označení kolik procent největších záznamů se má optimalizovat a následně vykreslit. Nabývá hodnot 0-100. Pokud je nastavena na 0, nevykreslí se žádný záznam, pokud je rovna hodnotě 100, vykreslí se všechny záznamy.

## 6.2.2 Případy optimalizace velkého počtu záznamů a vrcholů

Obecně může nastat pět případů, které ovlivňují způsob vykreslení a použití optimalizačních algoritmů:

### 1. Počet všech vrcholů je menší než MALO

Tento případ může nastat, jestliže je počet vrcholů nižší než hodnota proměnné **MALO**. Na mapovou komponentu se přidají i body, které jsou mimo aktuálně zobrazenou oblast. Bodů je nízký počet, a proto není rychlost vykreslení vrstev ovlivněna. Další výhodou přidáním všech vrcholů do mapy je to, že při posunu aktuálně zobrazené obrazovky na vrcholy, které byly dosud mimo oblast, jsou ihned vykresleny. Pokud tato podmínka není splněna, pokračujeme testováním druhé podmínky.

### 2. Počet všech vrcholů v zobrazené oblasti je menší než MALO

Tento případ nastane, pokud je souřadnic velké množství, ale v aktuálně zobrazené oblasti je jich jen pár (méně než **MALO**). Nyní budeme při vykreslování postupovat tak, jako bylo uvedeno výše (přidáme všechny vrcholy ležící uvnitř zobrazené oblasti a navíc vrcholy následující po změně regionu - bounding). Tím dosáhneme plynulé práce s mapou a korektním vykreslením všech polygonů (linií). Pokud tato podmínka není splněna, pokračujeme testováním třetí podmínky.

### 3. Počet záznamů je menší než MALO\_ZAZNAMU

Tato situace nastane, když máme příliš mnoho souřadnic uvnitř zobrazené obrazovky a menší počet záznamů (méně než **MALO\_ZAZNAMU**). Potom musíme provést více kroků k optimalizaci. Nejprve provedeme optimalizaci hraničních vrcholů (přidáme všechny vrcholy ležící uvnitř zobrazené oblasti a navíc vrcholy následující po změně regionu - bounding) a následně provedeme optimalizaci vrcholů ležících vně zobrazené obrazovky. K této optimalizaci budeme

muset vypočítat hodnotu relativní četnosti viditelných souřadnic daného záznamu z celkového množství viditelných souřadnic ve všech záznamech:

$$\text{relativni\_pocet} = \text{počet\_bodu\_v\_zaznamu} / \text{počet\_vsech\_viditelných\_bodu}$$

Nyní spočítáme, kolik bude vykreslených souřadnic, když celkem jich má být **MALO**:

$$\text{vykreslených\_souřadnic} = \text{relativni\_pocet} * \text{MALO}$$

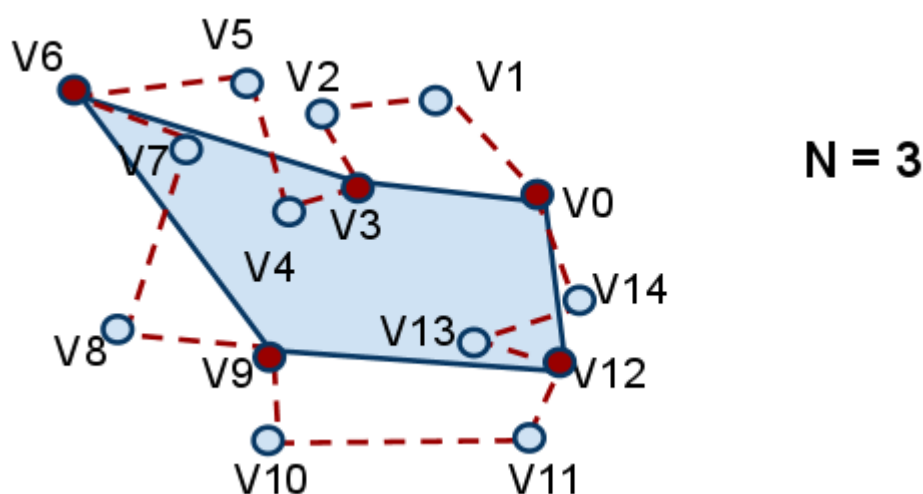
Pokud je hodnota **vykreslených\\_souřadnic** menší než hodnota **MINIMALNI\_ZAZNAM**, tak k výpočtu čísla **N** zvolíme výpočet:

$$N = \text{počet\_bodu\_v\_zaznamu} / \text{MINIMALNI\_ZAZNAM}$$

V opačném případě použijeme vzorec:

$$N = \text{počet\_bodu\_v\_zaznamu} / \text{Vykreslených\_souřadnic}$$

Díky těmto výpočtům získáme číslo **N**, které udává, kolikátá souřadnice z množiny všech viditelných souřadnic daného záznamu v pořadí se nemá vykreslit (Obrázek 25).

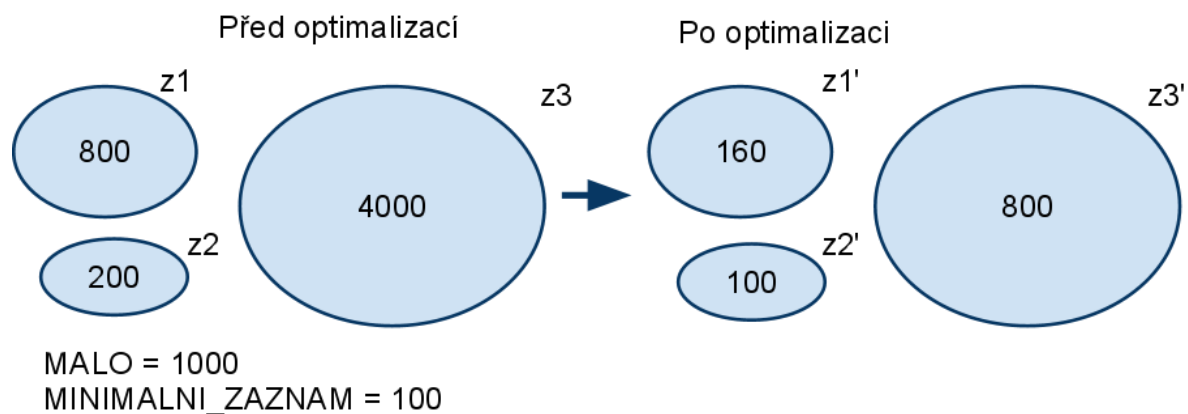


Obrázek 25: Zobrazení každé třetí souřadnice

Nyní počet zobrazených souřadnic se bude pohybovat okolo hodnoty **MALO**. Budou vykresleny všechny viditelné záznamy s tím, že záznamy, které by díky optimalizaci byly příliš zkreslené, jsou ovlivněny proměnnou **MINIMALNI\_ZAZNAM**. Ta nám například zaručí, že záznam s 200 souřadnicemi, který by mohl být díky nízké hodnotě **relativni\_pocet** minimalizován na 40 souřadnic, ale místo toho bude mít hodnotu **MINIMALNI\_ZAZNAM** (např. 100). Takže tvar bude lépe zachován.



Větší záznamy budou optimalizovány tak, že počet zobrazených souřadnic daného záznamu bude zachován ve stejném poměru k počtu zobrazených souřadnic ostatních záznamů (Obrázek 26).



Obrázek 26: Optimalizace počtu zobrazených bodů u záznamů

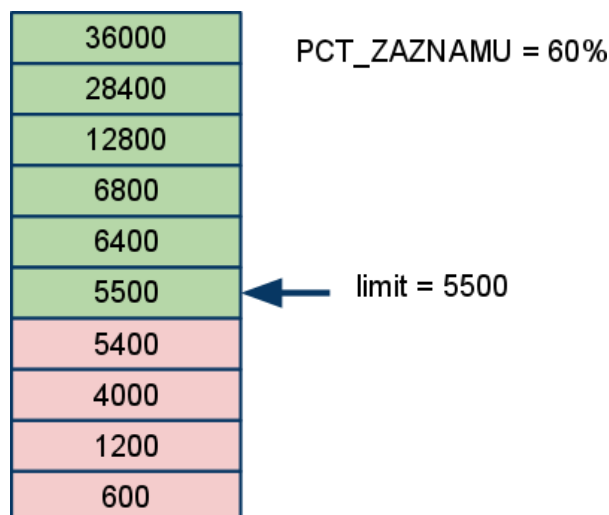
#### 4. Počet viditelných záznamů je menší než **HODNE\_ZAZNAMU**

Pokud máme velké množství viditelných záznamů, je vhodné nezobrazovat záznamy s menším počtem bodů. Ty by nám díky nízké hodnoty **relativni\_pocet** a vysoké **MINIMALNI\_ZAZNAM** mohly způsobit zpomalení aplikace, z důvodu přidání vyššího počtu vrcholů na mapu.

Nejprve je nutné určit hranici, které záznamy se mají vykreslit a které už ne. Tato hranice je dána hodnotou parametru **PCT\_ZAZNAMU**. Tento parametr u seřazeného seznamu s počty viditelných souřadnic jednotlivých záznamů určí limit, tj. minimální počet viditelných souřadnic.

Tento princip je znázorněn na Obrázek 27: Jednotlivá buňka tabulky znázorňuje počet viditelných souřadnic záznamu. Celá tabulka je seřazena sestupně. Je dán parametr **PCT\_ZAZNAMU = 60%**, čili 60% záznamů bude zobrazeno. Jsou to ty záznamy, které mají počet zobrazených souřadnic roven nebo vyšší **5500**.





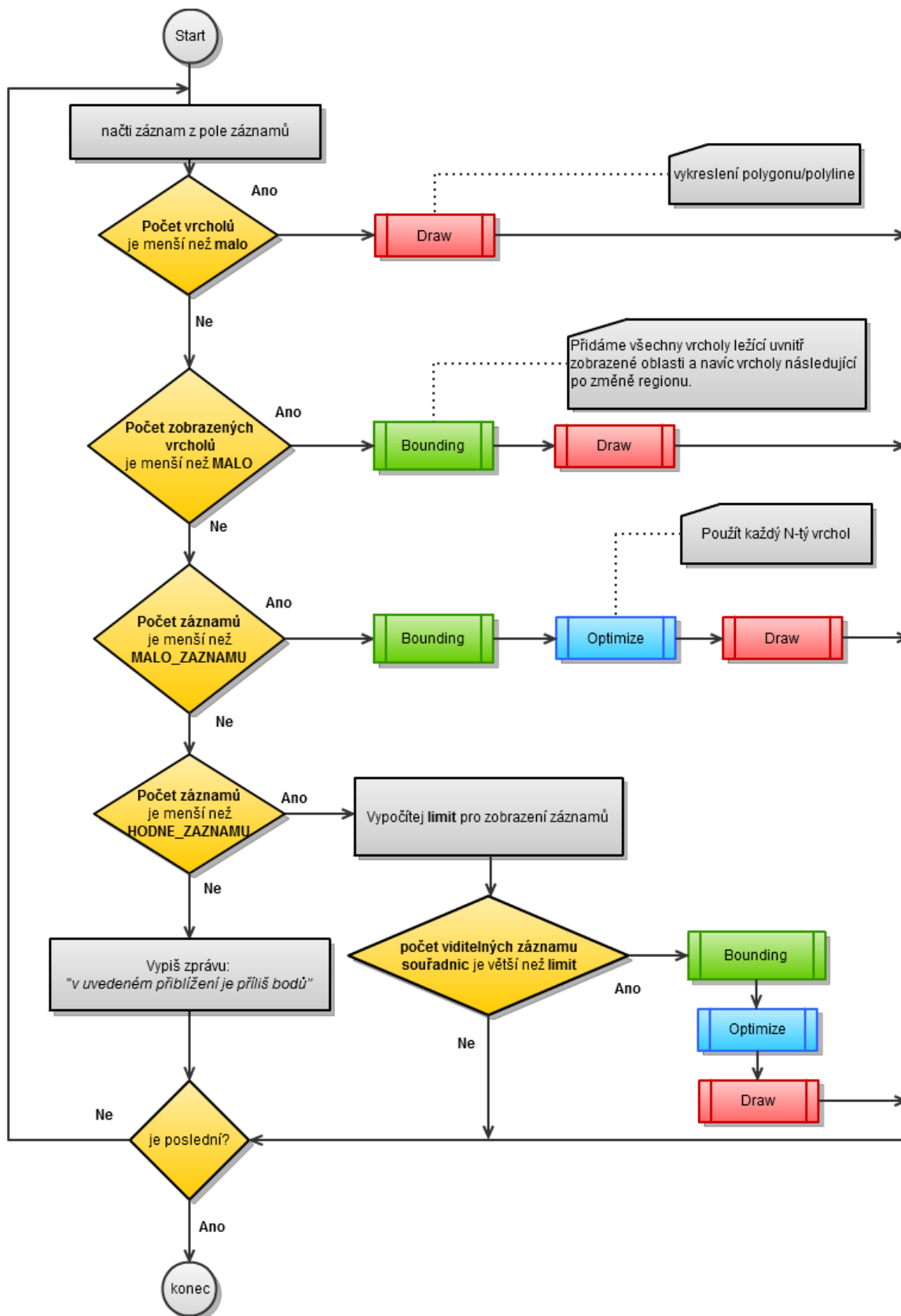
Obrázek 27: Limit zobrazených záznamů

Po vybrání záznamů, které se mají vykreslit, se opět provede optimalizace hraničních vrcholů (přidáme všechny vrcholy ležící uvnitř zobrazené oblasti a navíc vrcholy následující po změně regionu - bounding) a následně provedeme optimalizaci vrcholů ležících vně zobrazené obrazovky, jako to bylo uvedeno výše.

##### 5. Žádný z uvedených případů nenastal

V tomto bodě nám nezbývá nic jiného, než upozornit uživatele, že počet zobrazených záznamů je příliš vysoký na to, aby byla zachována vypovídací hodnota zobrazovaných informací. Je problematické vybrat záznamy, které se mají zobrazit. Na mapě by mohlo vzniknout příliš mnoho prázdných míst, které by následně mohly mást uživatele. Proto je vhodné jej upozornit, aby zvolil větší měřítko mapy na úroveň, kdy počet viditelných záznamů klesne na vhodnou mez.

Vývojový diagram případů optimalizace pro vykreslení je znázorněn na Obrázek 28.

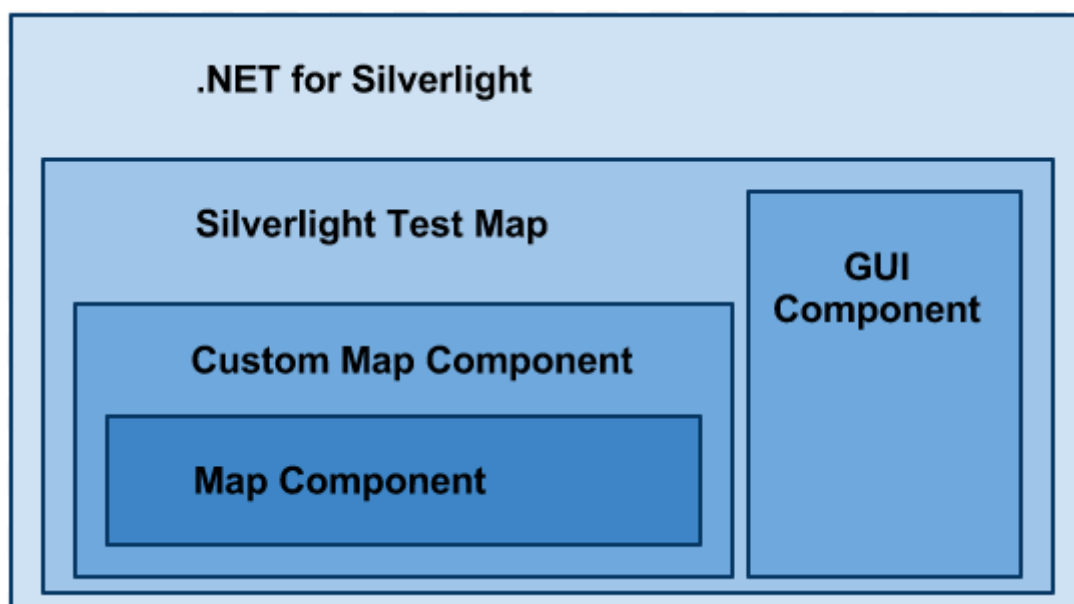


Obrázek 28: Vývojový diagram vykreslení a optimalizace

## 7. Struktura aplikace

V prostředí MS Silverlight (**.NET for Silverlight**) bude vytvořena testovací aplikace s názvem **Silverlight Test Map**. Ta bude obsahovat vlastní mapovou komponentu (**Custom Map Component**). Pro práci a testování mapové komponenty bude vytvořeno grafické uživatelské rozhraní. Při vývoji GUI může být využito možností komponent třetích stran (**GUI component**). Tyto komponenty rozšíří standardní funkcionalitu prostředí Silverlight o pohodlnější práci s mapovou komponentou. Vlastní mapová komponenta (**Custom Map Component**) bude postavena na mapové komponentě (**Map Component**) a rozšíří její funkcionalitu dle specifikace požadavků.

Na základě specifikace požadavků testovací aplikace je navržena struktura aplikace (Obrázek 29: Struktura aplikaceObrázek 29).



Obrázek 29: Struktura aplikace

## 8. Implementace

Tato fáze vývoje testovací aplikace je finální vývojovou etapou a vychází z předešlých fází. Zahrnuje vlastní programátorskou činnost. Výsledkem této vývojové fáze je funkční aplikace, která splňuje funkce, uvedené v počátku celého softwarového procesu – ve specifikaci požadavků.

K samotné implementaci testovací aplikace je využito vývojové prostředí Visual Studio 2008 SP1 a Silverlight Tools for Visual Studio 2008 SP1.

Pro běh aplikace je nutné mít nainstalovaný webový prohlížeč s rozšířením pro Silverlight. Konečná aplikace může být zprovozněna na libovolném webovém serveru, který podporuje práci s HTML. Silverlight aplikace je implementována jako vložený objekt ve webové stránce.

### 8.1 Technologie Silverlight

Silverlight je multi-platformní implementace firmy Microsoft, která dovolí návrhářům a vývojářům vyvinout Rich Internet Application (RIA) vloženou do webové stránky. Silverlight je vybaven flexibilní „rourou“, po které je velmi snadné přenést bohaté ovládací prvky do webových aplikací.

Nejčastějším frameworky nabízejícími RIA je Adobe Flash, Java a Microsoft Silverlight. Dle [3] Silverlight získává stále častější zastoupení na trhu.

Framework Silverlightu v sobě kombinuje tři velmi odlišné architektury: nástavba prohlížeče (browser plugin), prezentační framework a .NET framework. Spojení těchto architektur dovolí Silverlightu přemostit mezeru mezi uživatelským rozhraním použitím deklarativních jazyků a funkcí programováním podmnožinou .NET frameworku.

Jednoduchý browser plugin provádí nezbytnou interakci s prohlížečem a otevírá vrátka Silverlight aplikacím k běhu na mnoha platformách. Plugin musí být nainstalován uživatelem před tím, než může být Silverlight aplikace zobrazena v prohlížeči.[13]

Silverlight aplikace jsou implementovány jako vložený objekt do webové stránky. Když se prohlížeč setká s objektem Silverlight ve webové stránce, tak plugin stáhne balíček XAP z webového serveru, který obsahuje binární zdroje pro Silverlight aplikaci a potom bude kód spuštěn uvnitř webové stránky.

Silverlight aplikace běží jako client-side aplikace bez potřeby obnovení prohlížeče k aktualizaci uživatelského rozhraní (UI). Díky tomu, že Silverlight je postavený na .NET frameworku, Silverlight aplikace mohou jednoduše komunikovat s ovládacími prvky na straně serveru (server-side controls) a službami (service). Použitím implementace .NET frameworku v Silverlightu, vývojáři jednoduše mohou integrovat existující knihovny a kódy do Silverlight aplikace.

Prezentační framework Silverlightu je podmnožinou Window Presentation Foundation (WPF), která je založena na programovacím jazyce eXtensible Application Markup Language (XAML). Programovací jazyk XAML je jednoduše založený na XML jazyce s aplikačními elementy, které označují objekty a vlastnosti v .NET frameworku. Díky tomuto je možné syntaktickou analýzu (parsing) a integraci provádět spoustou různých technologií.

Silverlight framework umožňuje přistupovat k webovým službám, databázím, webovým serverům a dalším datovým zdrojům, ze kterých může čerpat data a zobrazovat je v uživatelském rozhraní aplikace. Také umožňuje aplikaci integrovat do webové stránky stejným způsobem jako AJAX a JavaScript rozšiřují funkcionalitu webové stránky. Navíc zaručuje uživateli, že aplikace bude vypadat a chovat se vždy stejně – nezávisí na webovém prohlížeči a platformě, na které běží.

#### Výhody prostředí Silverlight by se daly shrnout takto:

- **Multiplatformní** – je nezávislá na platformě – může běžet na více počítačových platformách (Windows, Linux, OS X)
- **Běží na více prohlížečích** – po instalaci jediného doplňku je možné zobrazit Silverlight aplikaci v jakémkoliv běžném internetovém prohlížeči
- **Vysoká kvalita** – Silverlight je vyvíjen pro novou generaci aplikací běžících v prostředí webového prohlížeče i mimo něj. Podporuje směřování toku médií (streaming) ve vysokém rozlišení – toho může být využito i ve Virtual Earth for Silverlight
- **Rychlý vývoj** – díky použití frameworku .NET, který je proslulý jako rychlé a efektivní vývojové prostředí
- **Vyšší výkon** – oproti JavaScriptových aplikací nabízí prostředí Silverlight vyšší výkon. Od Silverlightu verze 3 umožňuje přenést část zátěže procesoru na lokální GPU.

Platforma Silverlight obsahuje dvě hlavní části a zásuvný modul (plugin) viz Obrázek 30.

#### Prezentační jádro (presentation core)

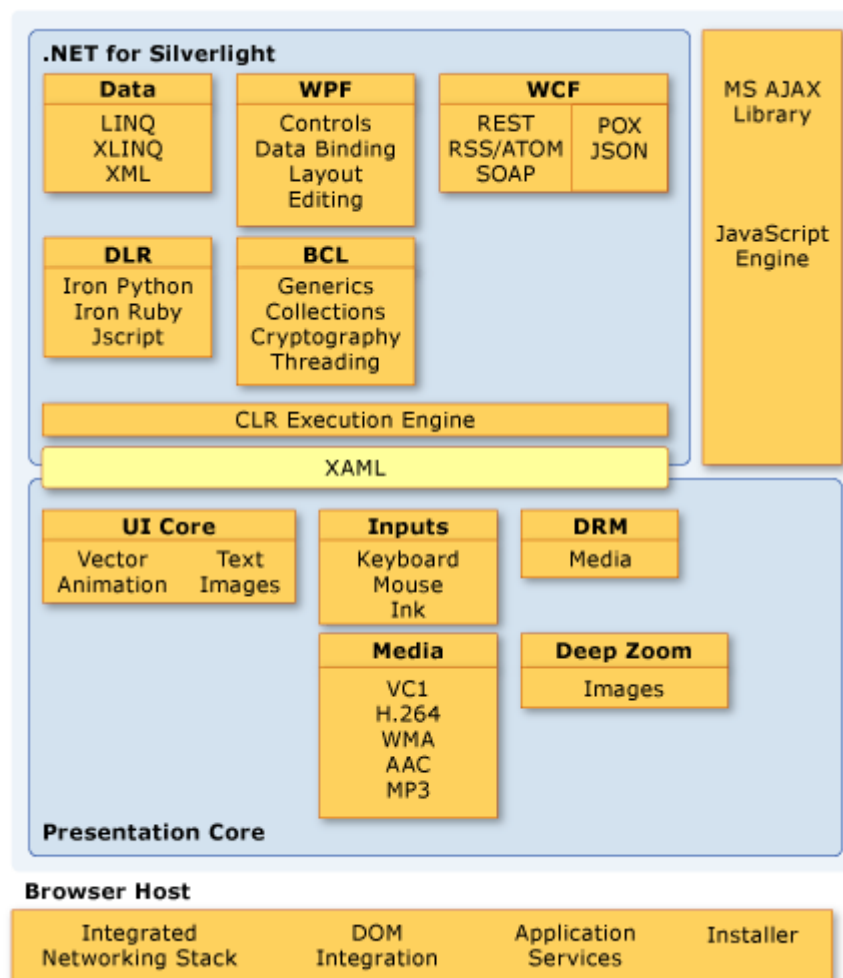
Prezentační jádro obsahuje komponenty a služby orientované na rozhraní a interakce s uživatelem. Lehká ovládací rozhraní pro použití ve webových aplikacích, přehrávání médií, správa digitálních práv, datovou vazbu. Dále obsahuje prezentační funkce zahrnující vektorovou grafiku, texty, animace a obrázky. Pro určení rozložení slouží XAML.

#### .NET for Silverlight

Podmnožina .NET frameworku, která obsahuje komponenty a knihovny včetně funkcí pro integrace dat a rozšířené ovládací prvky Windows. Některé části .NET frameworku jsou rozmístěny (deployed) v aplikaci. Tyto knihovny jsou dodávány v Silverlight SDK. Když jsou knihovny Silverlight použity v aplikaci, tak jsou přibaleny do XAP souboru a staženy do prohlížeče. Patří mezi ně nové ovládací rozhraní, XLINQ, Syndication (RSS/Atom), XML serializace a DLR.

#### Instalátor a aktualizací program (zásuvný modul)

Zjednodušují proces instalace aplikace pro nové uživatele a následovně provádějí automatické aktualizace.



Obrázek 30: Silverlight struktura

Zdroj: [http://msdn.microsoft.com/en-us/library/bb404713\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/bb404713(VS.95).aspx)

### 8.1.1 RadControls for Silverlight

Telerik je přední bulharská vývojová společnost zabývající se vývojem a prodejem ASP.NET AJAX, ASP.NET MVC, Silverlight, WinForms a WPF kontrol a komponent. Telerik pomáhá vývojářům vytvářet aplikace s nesrovnatelnou bohatostí a interaktivitou. Využití dalších komponent je velice efektivní a přinese do aplikace nové prvky, které by znamenaly spoustu času a usilí v případě, že by si je měl programátor vyvinout sám. Uživatelské rozhraní testovací aplikace je rozšířené právě o některé možnosti komponent RadControls for Silverlight.

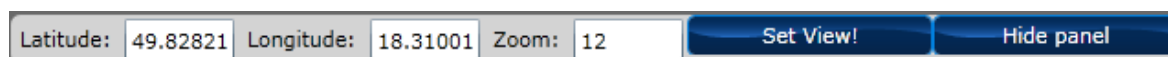
Hlavním přínosem komponent od firmy Telerik je využití funkce přetažení (drag&drop): V jedné části obrazovky jsou grafické ikony znázorňující datovou vrstvu a v druhé části obrazovky je panel obsahující ikony právě zobrazených datových vrstev. Mezi těmito panely je možné přesouvat ikony a zároveň zobrazovat/skrývat načtené vrstvy. Dalším přínosem komponent Telerik je nahrání datových souborů na server, které následně mohou být využity v mapové komponentě.

## 8.2 Uživatelské rozhraní

Testovací aplikace je navržena tak, aby snadno umožňovala nastavit všechny parametry, předvést funkcionalitu a práce s ní byla co nejvíce intuitivní. Na testovací aplikaci je prezentováno vše, co bude vlastní mapová komponenta umožňovat.

Testovací aplikace (Obrázek 32) se skládá z horní lišty, levého menu a panelu aktuálně načtených vrstev.

Horní lišta slouží k zadání souřadnic, míry přiblížení a následně umožní takto nastavit požadovaný pohled na mapu kliknutím na „Set View!“. Dále obsahuje tlačítko „Hide panel“ pro schování všech panelů – pro maximalizaci mapy (Obrázek 31).  
Obrázek 31: Horní panel testovací aplikace).



Obrázek 31: Horní panel testovací aplikace

Další částí je levé menu, které se skládá z panelu „All layers“. Ten obsahuje datové vrstvy, které jsou v aplikaci přístupné. Při prvním přihlášení se uloží uživateli seznam přístupných vrstev do IsolatedStorage. Uživatel testovací aplikace si po kliknutí na „New Layer from URL“ z panelu Menu může načíst vlastní datové vrstvy, které bude mít i při dalším spuštění testovací aplikace přístupné. Přístupné vrstvy z panelu „All layers“ může uživatel vymazat uchopením ikony a přetažením nad ikonu odpadkového koše „Recycle Bin“. V tomto případě se vymaže pouze záznam z IsolatedStorage. Data budou dále přístupná ze serveru dalším uživatelům. Panel „Menu“ dále obsahuje přepínače pro nastavení funkcí, které mají být aktivovány či deaktivovány. Dále umožňuje načíst novou vrstvu z URL (New Layer from URL) anebo přímo zobrazit soubor ShapeFile na mapu (Load ShapeFile). Panel „Map source“ umožňuje nastavit zdroj mapového podkladu.

Přístupné jsou tyto mapové podklady:

- Bing Aerial
- Bing Street
- Open Aerial
- Open Street
- Yahoo Aerial
- Yahoo Hybrid
- Yahoo Street
- None

Poslední část levého menu obsahuje panel „Optimize“. Zde je možné nastavit proměnné ovlivňující rychlost a detailnost vykreslených datových vrstev. Touto problematikou se podrobně zabývám v kapitole 6.2.1 a 6.2.1.



Panel aktuálně načtených vrstev (My Layers) umístěný v horní části obrazovky slouží k zobrazení ikon načtených vrstev nad mapovým podkladem. Tyto vrstvy je možné přidávat z panelu „All Layers“ uchopením ikony vrstvy a následným přetažením nad bílou plochu panelu „My Layers“. Zobrazenou vrstvu je možné skrýt přetažením její ikony do panelu „All Layers“, anebo odstraněním po přetažení nad „Recycle Bin“.



Obrázek 32: Testovací aplikace

### 8.3 Rozložení funkcí mapové komponenty

Samotná mapová komponenta obsahuje panely, které je možné libovolně zobrazit či skrýt (Obrázek 33). Panely je možné nastavit přes „Menu“ testovací aplikace, popř. přímo ze zdrojového kódu. Nastavit se dají tyto panely:

**Zobrazení lupy** – uživatel bude moci použít pro lepší orientaci na mapě mini-mapu. Jedná se o zmenšenou původní mapu s menším přiblížením, než které má hlavní mapa. Pozice mini-mapy se automaticky mění dle pohybu na hlavní mapě. Při přiblížení hlavní mapy se použije přiblížení na mini-mapě. Toto chování je umožněno i u mini-mapy - po posunu nebo použití zoomu na mini-mapě se změní aktuální pozice na mapě hlavní. Mini-mapu je možné zobrazit i skrýt.

**Zobrazení měřítka** – měřítko mapy se odvíjí od aktuálního přiblížení. Délka měřítka znázorňuje uvedenou vzdálenost.



**Zobrazení navigace** – jedná se o panel pro posouvání, přiblížení, oddálení mapy. S mapou je možné pracovat i bez použití tohoto panelu – pomocí uchopení myši, použitím šipek na klávesnici ap.

**Zobrazení souřadnic mapy** – obsahují informace o aktuální pozici na mapě: souřadnice X (zeměpisná délka), Y (zeměpisná šířka), přiblížení (zoom).

**Zprávy** – slouží k informování uživatele o provedených akcích. Jedná se o volitelný výpis informací na mapu.

**Omezení oblasti mapy na ČR** – při překročení hranic ČR se zobrazovaná oblast vrátí zpět na souřadnice ČR.



Obrázek 33: Panely na mapě

## 9. Budoucí nasazení, rozšíření, využití

Spektrum využití této komponenty může být velice široké, jelikož byla navržena pro podporu datových formátů, které mohou nést libovolná geografická data. Avšak předpokládá se, že hlavní využití komponenty bude v projektu FLOREON v částí, která se zabývá dopravní situací.

Tato komponenta nabízí prostor pro rozšíření dalších datových formátů, kterých je celá řada. Např. hojně využívaný datový formát ve spojení s GPS přístroji je GPX soubor, který umožňuje ukládat polohu v závislosti na čase – čili zaznamenávat trasu. K uložení polohy bez žádných zvláštních atributů může sloužit jednoduchý LOC soubor, o který by také bylo možné později mapovou komponentu rozšířit. Další rozšíření se nabízí u formátu KML, který může nést nepřeberné množství atributů. Zde hlavně záleží na atributových požadavcích a zpětné vazbě programátorů, kteří budou tuto komponentu využívat a dále s ní pracovat. Zpětná vazba je důležitá nejen u rozšíření komponenty o další funkcionalitu, ale také k otestování stávajících možností.

K těmto, ale i k dalším rozšířením je tato komponenta již připravena. Mapová komponenta může být využita i k práci s daty bez mapového podkladu. Příkladem takového projektu, ve kterém je raná verze mapové komponenty využita, je grafický vyhledávací nástroj IKE (I Know Everything), jehož smyslem je poskytnout přímočarou cestu k hledaným informacím.<sup>3</sup> IKE se dá definovat, jako vyhledávač vzájemně relativních klíčových slov.

Způsob práce s IKE lze zjednodušeně popsat takto: uživatel zadá klíčová slova (term) a IKE zobrazí seznam výsledků (rozumějte stránky, na kterých se tento term vyskytoval). Následně uživateli nabídne další slova, která s původně zadanými slovy souvisí. Každý term je graficky znázorněn jako kolečko (nebo chcete-li bublina či uzel). Jeho velikost a barva je znázorněna důležitostí termu v celkovém vzorku (Obrázek 34). Uživatel snadno pozná, jak velké množství informací lze o tomto termínu nalézt. Jednotlivé termy jsou propojeny hranami, jejichž tloušťka představuje množství odkazů (včetně počtu stránek, na kterých se vyskytují tyto termy současně) mezi těmito dvěma termy. Při kliku na některý z okrajových uzlů se i on rozvětví podobným způsobem atp. Grafické rozhraní tak realizuje spojitý graf, jehož nejdelší cesta popisuje historii vzájemně relativních termů, jimiž uživatel stále těsněji specifikoval předmět svého zájmu.[14]

IKE prohledává pouze omezenou množinu internetu, kterou tvoří česká wikipedia.

Systém je rozdělen na dvě samostatné komponenty:

- crawler (napsaný v technologii c# winforms)
- vyhledávač (napsaný v technologii Silverlight)

Crawler prochází data české wikipedia ve formátu XML. Cílem crawleru je indexovat slova nacházející se na jednotlivých stránkách a ukládat je ve vhodné podobě do souborů XML, který je vstupem pro vyhledávač. Při indexování je každému slovu přiřazována určitá důležitost v

---

<sup>3</sup> Systém IKE byl inspirován jedním z nejnovějších produktů společnosti Google, který se nazývá Wonder Wheel

závislosti na tom, kolikrát se na stránce vyskytuje a také speciálně za výskyt v nadpisech, odkazech a zvýrazněních.

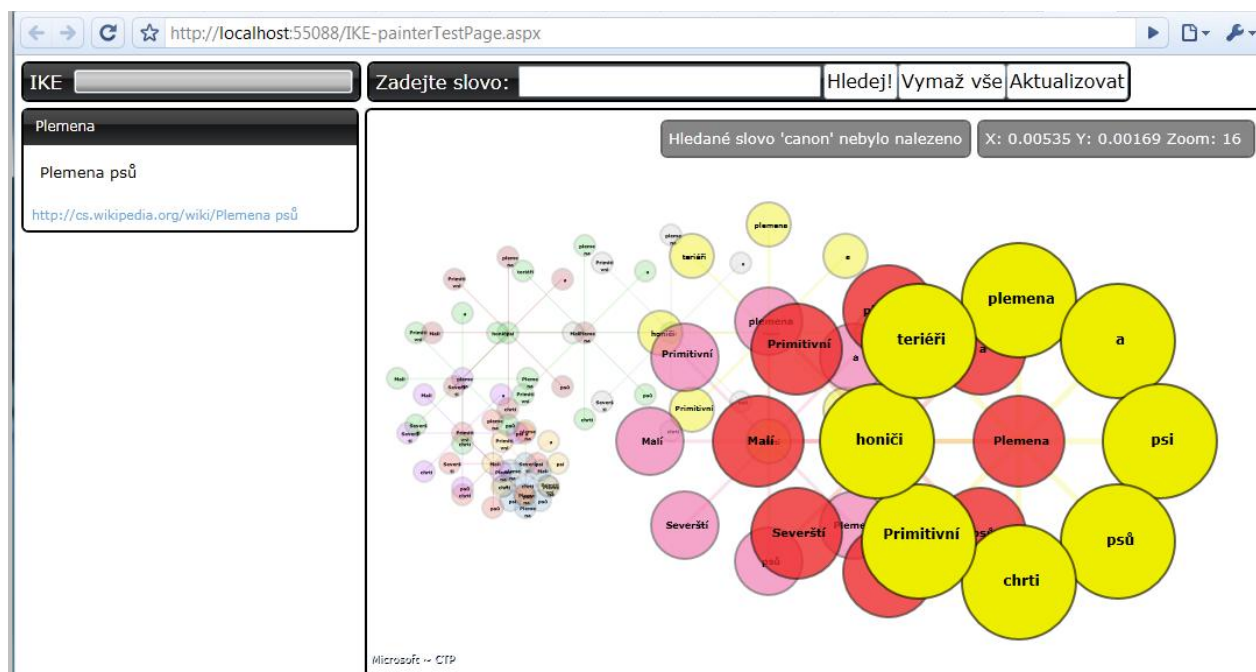
Cílem vyhledávače je realizovat grafické uživatelské rozhraní a vlastní funkce systému IKE. Vstupem pro něj budou XML soubory získané z crawleru.

Vyhledávač můžeme pomyslně rozdělit na:

- střední vrstvu
- uživatelské rozhraní

Hlavním úkolem střední vrstvy je vyhledávat požadované informace nad daty z crawleru a předzpracovávat je pro snadné vykreslení za pomoci Silverlight komponent. Vstupem pro střední vrstvu tedy jsou XML soubory z crawleru a výstupem XML soubor s daty pro vizualizaci, která následně postoupí do nadřazené vrstvy.

Uživatelské rozhraní je jediná část, se kterou uživatel přijde přímo do styku. Je realizováno prostřednictvím Silverlightu, který obsahuje komponenty pro zpracování vstupních XML a vykreslení grafů. Právě vykreslení grafu a zpracování XML je realizováno vlastní mapovou komponentou. Ta slouží k zasílání požadavků (reakce po kliknutí na bublinu obsahující klíčové slovo) a následné prezentování výsledků uživateli.



Obrázek 34: Nemapový projekt IKE

## 10. Závěr

Cílem této diplomové práce bylo zmapování možností vizualizace informací nad mapovým podkladem prostřednictvím technologie MS Silverlight. Následně jsem si vzal za úkol vytvořit testovací aplikaci, která bude ilustrovat možnosti integrace a využití mapové komponenty. Na základě zjištěných informací jsem dále měl vytvořit vlastní mapovou komponentu, která bude rozšiřovat její funkcionalitu především o možnosti práce s dalšími informačními vrstvami a formáty.

Tyto cíle jsem se snažil plnit v plném rozsahu. Navrhl jsem testovací aplikaci, která se stala stavebním kamenem pro vytvoření samostatné komponenty umožňující zobrazení dalších informačních vrstev. K této komponentě jsem sepsal instalační a programátorskou příručku. Tato dokumentace usnadní dalším programátorům integraci a využití komponenty v Silverlight aplikacích. Vytvořená komponenta již může již využita k implementaci, ale je zde určité riziko výskytu chyb. Testovací fázi jsem se zabýval sám pouze na lokálním vývojovém prostředí. V případě nasazení aplikace by pro mě byla zpětná vazba dalších programátorů velkým přínosem pro vyladění komponenty k dokonalosti.

V této diplomové práci jsem dospěl k závěru, že zobrazování informací na mapovém podkladu je stále častěji využívaným postupem, a proto si zaslouží patřičnou pozornost. Rozšíření funkcionality o zobrazení informací v závislosti na své poloze může mít vysoké možnosti využití a může ovlivnit životy každého z nás.

# Literatura

- [1] 1854 Broad Street cholera outbreak In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 10:12, 7 August 2008, 16:49, 21 February 2010 [cit. 2010-05-04]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/1854\\_Broad\\_Street\\_cholera\\_outbreak](http://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak)>.
- [2] *All About GIS* [online]. 15 Listopad 2009 [cit. 2010-05-04]. Nejenom co, ale také kde . Dostupné z WWW: <<http://www.allaboutgis.eu/index.php/gis/1-zakladni-informace/4-nejenom-co-ale-take-kde>>.
- [3] *StatOwl* [online]. 2010 [cit. 2010-03-28]. RIA Market Penetration and Global Usage. Dostupné z WWW: <[http://www.statowl.com/custom\\_ria\\_market\\_penetration.php](http://www.statowl.com/custom_ria_market_penetration.php)>.
- [4] *FLOREON* [online]. 20.4.2010 14:42:07 [cit. 2010-04-20]. Floreon+. Dostupné z WWW: <<http://floreon.vsb.cz/web/default.aspx>>.
- [5] *World Geodetic System* [online]. 2009 , 1. 12. 2009 v 18:22 [cit. 2010-03-22]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/World\\_Geodetic\\_System](http://cs.wikipedia.org/wiki/World_Geodetic_System)>.
- [6] Zeměpisná délka In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 24. 3. 2005, 12:39, 15. 3. 2010, 23:15 [cit. 2010-04-28]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Zeměpisná\\_délka](http://cs.wikipedia.org/wiki/Zeměpisná_délka)>.
- [7] *Via Windows Live* [online]. 2009 [cit. 2010-03-28]. Getting Started with the Virtual Earth Silverlight Map Control. Dostupné z WWW: <<http://www.viawindowslive.com/Articles/VirtualEarth/VirtualEarthSilverlightMapControl.aspx>>.
- [8] *Chris Pietschmann* [online]. 20. March 2009 [cit. 2010-04-18]. Bing Maps Silverlight CTP: Overlay OpenStreetMap, OpenAerialMap and Yahoo Map Imagery using Custom Tile Layers!. Dostupné z WWW: <<http://pietschsoft.com/post/2009/03/20/Virtual-Earth-Silverlight-Overlay-OpenStreetMap2c-OpenAerialMap-and-Yahoo-Map-Imagery-using-Custom-Tile-Layers!.aspx>>.
- [9] *ESRI Shapefile Technical Description* [online]. [s.l.] : [s.n.], 1998 [cit. 2010-04-10]. Dostupné z WWW: <<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>>.
- [10] Shapefile In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 26 September 2005, 19 March 2010 [cit. 2010-04-10]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Shapefile>>.
- [11] *All About GIS* [online]. 5 Listopad 2009 [cit. 2010-03-18]. Reprezentace prostoru pomocí vektorů . Dostupné z WWW: <<http://www.allaboutgis.eu/index.php/gis/1-zakladni-informace/14-reprezentace-prostoru-pomoci-vektor>>.
- [12] *Open Geo Spatial* [online]. 2008 [cit. 2010-04-28]. KML. Dostupné z WWW: <<http://www.opengeospatial.org/standards/kml/>>.
- [13] DAYLEY, Brad. *Silverlight 2 : Bible*. Indianapolis : Wiley Publishing, Inc., 2008. 558 s.
- [14] PYTLÍK, Tomáš; MAJER, Jan; VONDRÁČEK, Tomáš. *IKE : Projekt do předmětu Dokumentografické informační systémy*. Ostrava, 2009. 14 s. Semestrální práce. VŠB-TUO.

# Přílohy

- A. **Instalační příručka**, která je v elektronické podobě uložena na CD
- B. **Programátorská příručka**, která je v elektronické podobě uložena na CD
- C. **Disk CD**, obsahující elektronické verze dokumentů diplomové práce, zdrojové kódy apod.